CLUBE Z80

Janeiro/85

N.º 28

Coordenadores: MARIA IRENE • ALEXANDRE SOUSA • J. MAGALHÃES • AV. BOAVISTA, 832-2.º T. • TELEF. 65127 • 4100 PORTO

NESTE NÚMERO

No interior:

Folheto Mercado Z80

Edição: Clube Z80

Fotocomposição: Fotomecânica Mabreu/Porto

Impressão: Ramos dos Santos & C.ª, Lda./Porto

Tiragem: 500 exemplares, Janeiro 1985

EDITORIAL

OS COORDENADORES DO CLUBE Z80

Em Dezembro de 84 deixou a nossa companhia MARIA IRENE SANTOS, que foi impulsionadora de muito do que o Clube Z80 representa.
Por tudo o que produziu e pelo que deixou realizado, aqui fica o nosso abraço de despedida e o desejo de que as suas esperanças se concretizem em termos de realização profissional e em tudo o resto.
Sucede-lhe ANA PAULA CALADO com o mesmo entusiasmo com que vimos a sua antecessora «agarrar» o Clube e as suas tarefas que nem sempre são simples de executar.
Para a Ana Paula aqui fica expressa a nossa confiança e o nosso desejo de que tudo avance conforme o impulso que trazemos, ou seja, crescer e firmar a nossa existência.

Foram este mês distribuídos os primeiros textos de programas para adaptar e traduzir; entretanto, continuamos receptivos a mais ofertas de colaboração.
Uma palavra para o nosso amigo LUÍS MANUEL COSTA FILIPE/CALDAS DA RAINHA.
— Temos aqui um trabalho seu, e solicitamos que nos envie uma cassette gravada com o programa, dado que existe dificuldade em entender certas letras da sua listagem manuscrita; para nós é sempre fácil passar a cassette e listar a impressora.
Aproveitamos para pedir a todos os sócios que nos enviem colaboração: os programas devem chegar de preferência em cassette — nós prometemos que devolvemos as cassettes — e sempre acompanhadas de um DESCRITIVO que pode ser manuscrito.

1985 e o CLUBE Z80

1) ZIG-ZAG (a diferença entre o útil e o acessório)

A ideia do Hugo Assumpção lançada em Novembro e que consistiu no desafio, em termos de apresentar um problema de programação, e publicar as respostas, vem ao encontro de muitas que tem sido enviadas ao Clube Z 80.

DESAFIO: PARA CRIAR UM ESPÍRITO DE DESCOBERTA DAS SUAS POTENCIALIDADES INVENTIVAS AO NÍVEL DA PROGRAMAÇÃO E LEVÁ-LO A USAR MAIS A CABEÇA E O LÁPIS DO QUE OS DEDOS E OS REFLEXOS, PROPÕE-SE ESTE DESAFIO:

CRIAR UM PROBLEMA QUE DEVE SER RESOLVIDO COM UM PROGRAMA EM BASIC OU PASCAL.

Todos nós assistimos ao desenrolar de uma série de provas, passadas nas tardes de domingo, e a que se chamava Campeonato de Computadores.
Pessoalmente comparo aquilo que observei, a uma série de provas de destreza, que nada tem a ver com o computador, a não ser que este é o suporte mecânico e electrónico da prova.
Foram publicadas páginas inteiras, na imprensa diária e semanal, sobre os «Campeões», e nunca o jornalista, foi capaz de discernir, entre o que deve ser o despertar do jovem (e não só) para a informática e a utilização do microcomputador como objecto lúdico ou seja de diversão.
Não somos contra o uso do Spectrum ou de outra máquina, como meio de diversão e até de desenvolvimento dos reflexos e da capacidade de observação.
Admiramos muitas vezes, a espantosa imaginação de autores famosos, que lançaram alguns dos mais famosos jogos e simulações que por aí circulam.
No entanto, não podemos aprovar a mistificação e o encobrir pelo silêncio, o desvio dado à utilização da máquina, em vez de a aplicar também, noutros campos.

Agarrando na ideia do HUGO ASSUMPÇÃO, poderíamos ampliar essa ideia e abrir uma «CAIXA», para a compra de dois Spectrum's. Os sócios fariam ofertas para essa caixa, a partir de um mínimo de 100$00.
O total amealhado, seria aplicado na atribuição de dois prémios para um concurso do seguinte tipo:

a) Aberto a grupos escolares, desde que o responsável seja um professor. Poderão existir mais do que um grupo por escola.
b) O concurso envolve um prémio para o melhor problema, colocado entre Dezembro e Abril, e outro prémio para a melhor solução remetida entre Janeiro e Maio, e aplicada a um dos problemas publicados.
c) Os sócios do Clube Z80 votam individualmente os problemas de programação e as soluções apresentadas, remetendo uma classificação de 1 a 5 para os casos publicados.

No final de Maio ou início de Junho, o problema e a solução mais votados, serão premiados com um Spectrum cada.
Em princípio sugerimos que o Spectrum seja entregue ao grupo organizado na escola e não ao autor individual.
Ficamos desde já com a «CAIXA» aberta às vossas ofertas e às vossas sugestões, problemas e soluções.

PROBLEMA DE JANEIRO DE 1985

Colocado pela Escola Secundária de SILVES
Alunos do 11.º ano — A1/A2

Fazer um programa para resolver equações do 2.º grau e que dê soluções da forma com que habitualmente se apresentam nos «cadernos».
Assim deverá apresentar soluções do tipo:

$x = \frac{2+\sqrt{3}}{5}$ ou $x = \frac{4}{7}$ e não na forma de dízima

INTRODUÇÃO À LINGUAGEM MÁQUINA

ZX81/SPECTRUM

Autor: FERNANDO PRECES
SACAVÉM

(Cont. dos números anteriores)

```
           10101010
              OR
           11000000
Resultado  11101010
```

Mnemónicas	Códigos
OR + N	246, + N
OR A	183
OR H	180
OR L	181
OR B	176
OR C	177
OR D	178
OR E	179
OR (HL)	182
OR (IX + d)	221, 182, + d
OR (IY + d)	253, 182, + d

O exemplo seguinte, extraído do programa monitor do ZX81, mostra-nos como é usada a instrução OR + N como **(set)** mascarador **1** dos bits 5, 6 e 7, de um byte contido no registro A.
A rotina de descodificação do Teclado, permite que este seja pesquisado por 8 varrimentos consecutivos cada 1/50 do segundo.
Cada varrimento é efectuado numa sequência diferente, a fim de tornar **exacta** a localização da tecla premida.
Como porém apenas são necessários 5 dos 8 bits, por cada um dos oito varrimentos do teclado, torna-se indispensável assegurar que os 3 bits restantes sejam mantidos ao mesmo valor. É aí que vai encontrar a operação de mascaramento.

Endereço	Código	Mnemónica	Comentários
709/0	246,224	OR + 224	Mascara a 1, os bits 5, 6, e 7

Exemplo da operação:

```
   101  10101
       OR
   111  00000
   111  10101
   ---
bits mascarados
```

Para o Spectrum escolhemos o seguinte ensaio:

Uma mensagem secreta é colocada na REM 1, pela rotina (linhas 20 a 80).
Após escrita, a mensagem é mascarada pelo trabalho da rotina (linhas 100 a 150).
Um descodificador que fará aparecer de novo a mensagem é executado pela rotina (linhas 200 a 250).
As duas rotinas em C/M., são colocadas no endereço escolhido pelas (linhas 500 a 600).

O espaço reservado do Buffer da impressora (quando esta se não encontra em funcionamento), pode ser utilizada para albergar pequenas rotinas em código máquina, que não excedam 256 bytes.

Área do Buffer da impressora
23296 a 23551 = 256 bytes.

1.ª Rotina (mascaramento mensagem).
Esta rotina mascara cada um dos bytes da mensagem, auxiliada por um contador em Basic.

Endereços	Mnemónicas
23300/2	LD A, (23728)
23303/4	OR + 128
23305/7	LD (23728), A
23308	RET

2.º Rotina (descodificação da mensagem)
Esta rotina limpa cada um dos bytes da mensagem, auxiliada por um contador em Basic.

Endereços	Mnemónicas
23320/2	LD A, (23728)
23323/4	AND + 127
23325/7	LD (23728), A
23328	RET

Programa em Basic

1 REM (reserve quantos caracteres forem necessários para a mensagem).

```
 20 INPUT "ESCREVA A MENSAGEM SECRETA"; a$
 30 LET X = 23760
 40 FOR N = X - TO LEW 9$
 50 POKE X, CODE a$ (N)
 60 LET X = X + 1
 70 NEXT N
 80 LIST
 99 REM * MASCARA * GOTO 100
100 LET X = 23760 : FOR N = X TO X + LEN a$
110 POKE 23728, PEEK N
120 RANDOMIZE USR 23300
130 POKE N, PEEK 23728
140 NEXT N
150 CLS : LIST : PAUSE 0
199 REM * LIMPEZA * GOTO 200
200 LET X = 23760 : FOR N = X TO X + LEN a$
210 POKE 23728, PEEK N
220 RANDOMIZE USR 23320
230 POKE N, PEEK 23728
240 NEXT N
499 REM * INTRODUÇÃO DO C.M. * GOTO 500
```

```
500 LETX Z = 23300
510 RESTORE 600
520 READ a : POKE Z, a
530 PRINT Z, PEEK Z
540 LET Z = Z + 1
550 GOTO 520
600 DATA 58, 176, 92, 246, 128, 50, 176, 92, 201, 0, 0, 0, 0,
    0, 0, 0, 0, 0, 0, 0, 58, 176, 92, 230, 127, 50, 176, 92, 201.
```

Subgrupo C — **As instruções XOR**

A operação lógica XOR permite testar cada um dos 8 bits do registro A com o bit correspondente quer dum registro, quer dum conteúdo de memória, modificando de seguida, face ao resultado, cada bit que sofreu alteração.
A operação XOR **especifica** que o bit resultante do teste seja colocado a **1** se um (e não ambos) dos bits testados for **1**. Nos restantes casos o bit é colocado a **0**.

Exemplo:

10101010	Conteúdo de A
XOR	
11000000	
01101010	Resultado para A

A operação XOR A, é uma das mais utilizada deste subgrupo, pois a sua execução **limpa** (coloca a zero) o **registro A e o carry flag.**

Mnemónicas	Códigos
XOR + N	238, + N
XOR A	175
XOR H	172
XOR L	173
XOR B	168
XOR C	169
XOR D	170
XOR E	171
XOR (HL)	174
XOR (IX + d)	221, 174, + d
XOR (IY + d)	253, 174, + d

Podemos aproveitar o programa anterior, «ESCREVA UMA MENSAGEM SECRETA», para ensaiarmos as instruções XOR.
Para tal vamos introduzir nesse ensaio, as seguintes alterações:

LINHA 99 REM * MASCÁRA OU LIMPA * GOTO 100

Apagar as linhas 199 a 240

Linha 600 data 58, 176, 92, 238, 128, 50, 176, 92, 201

Nota importante: Todas as instruções lógicas deste grupo (AND, OR e XOR) limpam (colocam a zero) o carry flag.

Com estas instruções, acabámos a parte **passiva** do nosso texto. Ao iniciar a explicação dos grupos seguintes, poderemos enveredar por um maior aproveitamento do código máquina, **excluindo** dos nossos ensaios os muitos contadores em Basic, que irão ser substituídos por instruções de salto e contadores muito mais rápidos.

Grupo 10 — **As instruções de salto**

No Basic dispomos de vários comandos que permitem alterar a sequência normal na execução dum programa, proporcionando um **Salto** para uma outra sequência de instruções (GOTO), para um subrotina (GOSUB) ou ainda, estabelecendo um ponto de decisão (IF seguido de THEN GOTO ou GOSUB).
O assembler possui instruções um tanto ou quanto semelhantes, que podem ou não estar sujeitas à condição do **estado dum flag.**
Estas instruções são muito importantes, pelo que para as estudar em pormenor, vamos dividi-las em 7 subgrupos distintos.

Subgrupo A — **Instrução de Salto Absoluto**

Esta instrução é directamente equivalente ao comando Basic GOTO, visto que unicamente indica ao Z80 o novo endereço para o registro Contador de Programa.
O registro PC (do qual já foi feita uma referência no início deste capítulo), contém sempre o endereço da instrução em execução e actua como apontador ao **serviço exclusivo do Z80.**
Ao alterarmos o seu conteúdo, através da instrução de salto, é executado de imediato a comutação para o novo endereço.

Mnemónicas	Código
JP addr	195, N, N

Em que o primeiro N imediatamente a seguir ao código da instrução é o LOW BYTE e o seguinte o HIGHT BYTE.

Tempo de execução:

N.º de Bytes	N.º Ciclos M	N.º Ciclos T
3	3	10

Esta instrução em **endereço absoluto** é uma ordem directa que, como é óbvio, não está dependente do estado dos flags.
Para saltos inferiores a **128 endereços** nas duas direcções possíveis (recuando ou avançando ao longo dos endereços de memória), utiliza-se uma outra instrução de salto, cujo tempo de execução é levemente maior, mas tem a vantagem de usar menos 1 byte.

Subgrupo B — **Instruções de salto com endereçamento indirecto**

Já sabemos que este tipo de endereço é uma espécie de segunda morada, que se encontra dentro duma caixa cujo endereço se conhece. Assim, por exemplo, se tivermos o registro HL a **apontar** o endereço 40000, cujo conteúdo é outro endereço, nós dizemos que HL contém um **endereço indirecto.**
Muitos programas comerciais, para se tornar difícil de descobrir aonde começam certas rotinas que executam coisas es-

pectaculares, repartem o programa de iniciação por vários pontos da memória, colocando num desses a carga do registro HL com o endereço da rotina principal e por fim a instrução de salto JP (HL). Outros carregam o Stack (Pilha) com vários endereços e depois de outros tantos saltos e muita confusão (para baralhar o próximo), o salto JP (HL).
Enfim, tendo em conta que se variarmos um endereço dum registro podemos alterar o endereço de um salto, tal como no Basic um GOTO n (em que n é uma variável) então é certo que muita coisa pode ser feita com estas instruções.
Este tipo de instrução não é afectado nem afecta os flags, executando apenas a carga do contador de Programa com o respectivo endereço.

Mnemónicas	Códigos	Tempos
JP (HL)	233	(a)
JP (IX)	221, 233	(b)
JP (IY)	253, 233	('')

Tempos de execução:

	N.º de Bytes	N.º Ciclos M	N.º Estados T
a	1	1	4
b	2	2	8

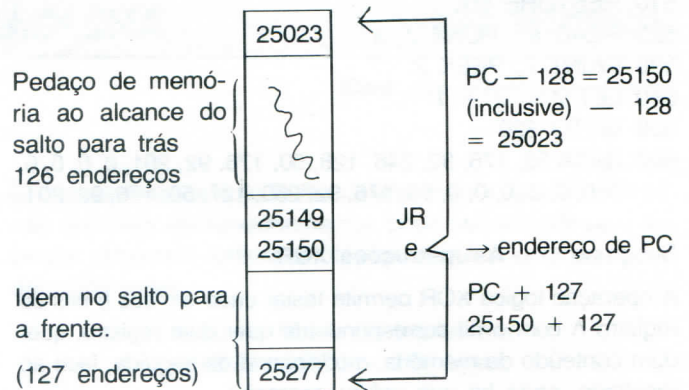
Subgrupo C — **Instrução de salto relativo**

Esta instrução é muito utilizada, visto que apenas 2 bytes são necessários para a sua formação, sendo o primeiro o o código da instrução e o segundo um endereço formulado em 2.º complemento aritmético.
Os números compreendidos entre 255 e 128, provocam saltos para traz (por exemplo, o número 200 executa um salto de 56 endereços no sentido descendente da memória, 200 − 256 = − 56).
Os números compreendidos entre 1 e 127 executam saltos para a frente.
Com este tipo de instrução o registro PC não recebe um novo endereço. Ele aceita o byte em 2.º complemento aritmético (também chamdo complemento por 2) para o **somar algebricamente ao endereço presente.**
Usando instruções de salto 'JR, é' (em que 'e' representa o byte em complemento por 2) torna-se possível alterar a posição duma rotina na memória, desde que não existam instruções de salto em endereço absoluto. É o caso dos programas ditos **recolocáveis** de que falaremos mais à frente.

Mnemónica	Código
JR e	24, e

Tempo de execução:

N.º de Bytes	N.º Ciclos M	N.º Estados T
2	3	12

Exemplo:

Se o leitor tiver alguma dificuldade em compreender o processo, deve seguir com muita atenção os ensaios que mais adiante serão apresentados para prática das instruções de salto.

Subgrupo D — **Instruções de salto condicionado ao estado do CARRY FLAG.**

São 4 as instruções que permitem saltos relacionados com o estado deste flag.
À semelhança do que já fizemos com outras instruções, poderemos afirmar que estas são equivalentes aos comando Basic:

IF C **THEN GOTO...** (se carry = 1)
IF NOT C **THEN GOTO...** (se carry = 0)

As duas primeiras instruções (JP) permitem saltos em endereçamento absoluto e as restantes (JR), saltos relativos.

Mnemónicas	Códigos	Comentários
JP NC , addr.	210, N, N	Salto se carry = 0
JP C , addr.	218, N, N	Salto se carry = 1
JR NC , e	48, e	Salto se carry = 0
JR C , e	56, e	Salto se carry = 1

Todas as instruções condicionadas ao estado dum flag, têm 2 tempos diferentes de execução.

a) quando a condição se não verifica — menor tempo
b) quando a condição se verifica — tempo indicado

Tempos de execução:

Tipo	N.º de Bytes	N.º Ciclos M	N.º Estados T
JP	3	3	10
JR	2	3	12

Subgrupo E — **Instruções de salto condicionado ao estado do flag 'ZERO'**

As 4 instruções deste subgrupo são equivalentes aos comando Basic:

IF Z **THEN GOTO...** (se z = 1)
IF NOT Z **THEN GOTO...** (se z = 0)

São 2 instruções para saltos absolutos (JP) e 2 para saltos relativos (JR).

Mnemónicas	Códigos	Comentários
JP NZ , addr	194, N, N	Salto com 'ZERO' = 0
JP Z , addr	202, N, N	Salto com 'ZERO' = 1
JR NZ , e	32, e	Salto com 'ZERO' = 0
JR Z , e	40, e	Salto com 'ZERO' = 1

Os tempos de execução são idênticos aos apresentados no subgrupo anterior.
Em conjunto com as instruções de comparação estes **saltos** têm uma larga aplicação em código máquina.

Subgrupo F — **Instruões de salto condicionado ao estado do flag 'SINAL'**

Há duas instruções condicionadas ao estado 'positivo' ou 'negativo' deste flag, que estão relacionadas com os códigos em complemento por 2. São instruções pouco utilizadas e quase sempre evitáveis.

Mnemónicas	Códigos	Comentários
JP P , addr.	242, N, N	Salto se 'SINAL' = 0
JP M , addr.	250, N, N	Salto se 'SINAL' = 1

Subgrupo G — **Instruções de salto condicionado ao estado do flag 'FORA DE ESCALA/PARIDADE**

São 2 as instruções de salto condicionado a este flag.

Mnemónicas	Códigos	Comentários
JP PO , addr.	226, N, N	Salto se 'P/O' = 0
JP PE , addr.	234, N, N	Salto se 'P/O' = 1

Apesar de menos utilizadas na chamada programação vulgar, as instruções pertencentes aos 2 últimos subgrupos são importantes no trabalho com algumas instruções mais complexas do Z80. Assim, voltaremos a falar nelas com detalhe na devida altura.

Ensaios práticos

Sempre que em C/M, temos de passar por cima duma subrotina, quadro de dados, etc., utilizam-se instruções de salto tipo JP e JR.

Ensaio 1: Salto absoluto sobre 1995 endereços

RAMTOP em 29999

Endereços	Mnemónicas	Códigos	Comentários
30000	LD A, 50	62,50	
30002	LD E, 70	30,70	
30004	ADD A, E	131	
30005	**JP 32000**	195,0,125	Salto com Endereço absoluto
32000	LD B, 0	6,0	
32002	LD C, A	79	
32003	RET	201	

PRINT USR 30000

Ensaio 2: Demonstração de instruções de salto absoluto e relativo.

Endereços	Mnemónicas	Códigos	Comentários
30000	**JP 32000**	195,0,125	Salto absoluto
30050	LD A, 50	62,50	
30052	**JR, + 9**	24, 9	Salto de + 10 endereços
30063	LD E, 70	30,70	
30065	ADD A,E	131	
30066	**JR, 188**	24,188	Salto de — 68 endereços
32000	LD B, 0	6,0	
32002	LD C, A	79	
32003	RET	201	

PRINT USR 30050

Ensaio 3: Salto condicionado ao estado do carry flag.

Endereços	Mnemónicas	Códigos	Comentários
30000	—	—	Valor a introduzir
30001	LD A, (30000)	58,48,117	
30004	LD E, 100	30,100	
30006	ADC A, E	139	
30007	LD C, A	79	
30008	LD B, 0	6,0	
30010	**JR C, + 1**	56,1	Salto se carry = 1
30012	RET	201	
30013	INC B	4	
30014	RET	201	

POKE 30000, N (N um número entre 0 e 255)
PRINT USR 30001

Ensaio 4: Salto condicionado do estado do zero flag. (Este ensaio destina-se apenas ao Spectrum)

Endereços	Mnemónicas	Códigos	Comentários
30000	LD B, 32	6,32	Uma linha de écran
30002	LDHL, 16704	33,64,65	Aponta a zona de écran
30005	LD A, 204	62,204	Formato do pixel
30007	LD (HL), A	119	Introdução
30008	INC HL	35	Endereço seguinte
30009	DEC B	5	Um a menos no contador
30010	LD A, B	120	Contador em A
30011	CP, 0	254,0	Compara com zero
30013	**JR NZ, _ 10**	32,246	Repete se A < > 0
30015	RET	201	

RANDOMIZE USR 30000

Ensaio 5: Idem, para écran completo.

Endereços	Mnemónicas	Códigos	Comentários
30000	LD BC, 6144	1,0,24	Para 6144 pixeis
30003	LD HL, 16384	33,0,64	Apontador (écran)
30006	LD A, 204	62,204	Definição do formato
30008	LD (HL), A	119	Transferência
30009	INC HL	35	Os 8 bits seguintes

30010	DEC BC	11	Contador — 1
30011	LD A, B	120	H byte da contagem em A
30012	OR C	177	OR com **L** byte
30013	CP, 0	254,0	Compara com 0
30015	JR NZ, — 11	32,245	Repete se A < > 0
30017	RET	201	

RANDOMIZE USR 30000

Ensaio 6: (destina-se ao ZX81) salto condicionado à flag zero. Simula o flash do Spectrum e a rotina pode ser toda montada um código máquina.

Endereços	Mnemónicas	Códigos	Comentários
16514	—	N	; Variável depende extensão da frase
16515	LD HL (16398)	42,14,64	; Pos. Print no Ficheiro écran
16518	LD A (16514)	58,130,64	; Lê a variável
16521	LD B, A	71	; Passa o valor para o contador
16522	LD A, (HL)	126	; Endereço do caracter
16523	XOR 128	238,128	; Inversão de vídeo do caracter
16525	LD (HL), A	119	; Devolve o caracter
16526	INC HL		; Próximo caracter
16527	DEC B		; Contador — 1
16528	LD A, B	120	; Contador em A
16529	CP, 0	254, 0	; Compara com 0
16531	JR NZ, — 11	32,245	; Ciclo, repete se A < >0
16533	RET		

Basic:

```
100 LET X = 16515
110 INPUT A$      → (introdução do texto)
120 LET B = LEN A$
130 LET Z = X − 1
140 POKE Z, B
150 PRINT AT 10, 15 − (B/2); A$
160 FOR N = 0 TO 100
170 FOR M = 0 TO 5
180 NEXT M
190 PRINT AT 10, 15 − (B/2);
200 RAND USR X
210 FOR M = 0 TO 5
220 NEXT M
230 NEXT N
```

NOTA: Todas as linhas em Basic, excepto as linhas 110 INPUT A$ e 200 RAND USR X podem ser transformadas também em código máquina.

(Continua no próximo número)

HARDWARE

Autor: Alexandre Sousa

256 valores são suficientes para representar todos os caracteres do alfabeto: maiúsculo; minúsculo; sinais de pontuação e caracteres especiais.

A comunicação entre dispositivos é possível, apenas com 8 linhas desde que tanto o Receptor como o Emissor concordem com cada um dos 256 valores representados. A segunda tarefa da unidade I/0 é precisamente a de assegurar esse acordo entre os dois dispositivos que comunicam entre si ou, no mínimo, tem de converter um conjunto de valores noutro conjunto que seja aceite pelo dispositivo.

Nem todos os dispositivos comunicam com o mesmo número de linhas. Alguns usam um simples condutor (mais a ligação de Massa-GND-) e enviam apenas um bit (dígito binário) de informação de cada vez.

O dispositivo que recebe a mensagem «reassume» esta informação de bits sequenciais numa representação «paralela» (por exemplo, 8 bits de dados guardados em oito linhas paralelas de dados).

Outros dispositivos enviam apenas numerais que podem ser representados por 10 valores e requerem apenas quatro condutores para sinais digitais (observe que 1010 em código binário, tem 4 bits e significa decimal 10).

Outras formas de representação podem exigir 16, 24, 32 ou 64 linhas, o que são já ligações complicadas.

Temos vindo a desenvolver alguns tópicos relacionados com o computador e a unidade I/0. Veremos em seguida como podemos dar ao programador, meios para que ele questione o computador e como pode o computador responder a essas questões.

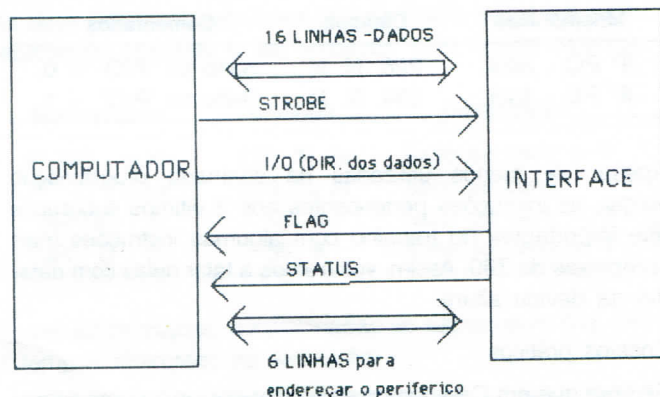



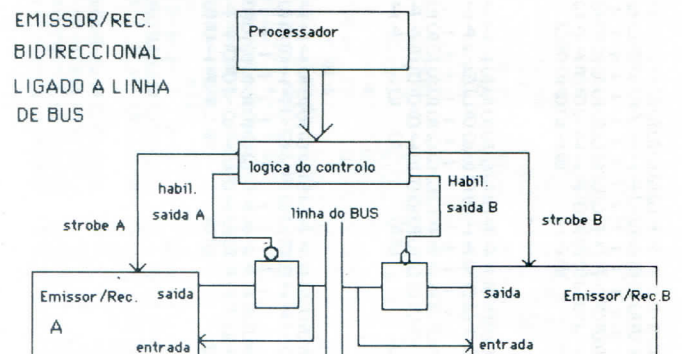
FIGURA 1

O primeiro passo será o de criar um BUS entre o processador e o mundo que o rodeia. Como vimos anteriormente, um BUS de Entrada/Saída é um conjunto de fios condutores que permitem a passagem de sinais representando toda a informação que o computador transmite e recebe — desde o processador até ao periférico.

Além desses, existem condutores que são portadores de sinais de controlo, geralmente para confirmar que os dados presentes no BUS são válidos e que podem ser aceites pelo dispositivo receptor.

O periférico comunica que os dados foram lidos e aceites, enviando um sinal de controlo que habilita o computador a receber e enviar sinais e ditar qual a direcção do fluxo de dados no interior do I/0 BUS.

Podem observar na figura 1 que a ligação superior, com flechas nas extremidades, representa 16 linhas de dados. A comunicação estabelece-se em ambos os sentidos e esse sentido depende apenas da necessidade imediata do processador.
O responsável pelo sincronismo, ou seja, pela 'bandeira' de partida dos dados, é um simples condutor, denominado STROBE.
O computador usa o Strobe para indicar que os dados estão disponíveis — prontos a ser aceites.
O próximo condutor chama-se I/0 Bus, permitindo que o fluxo bidireccional de dados circule num só sentido de cada vez.
Os sinais que provêm do periférico chamam-se STATUS e FLAG.
O Status é um simples sinal indicando ausência ou presença do periférico (no sentido de estar ou não presente para receber dados).
Não esquecer que o computador não pode comunicar com um periférico que não esteja presente.
FLAG é um sinal complexo. Para o compreender terá de relacioná-lo com a velocidade. Os processadores são muito rápidos, as únicas partes móveis no seu interior são electrões como portadores de sinais digitais.
Por outro lado, muitos periféricos com os quais o computador comunica, são do tipo mecânico.
Discos; impressoras; leitores de cassettes e plotters, todos são possuidores de partes mecânicas que se movimentam lentamente, consumindo longos períodos de tempo para executar qualquer tarefa.
Observem uma impressora, por exemplo, e reparem no processo de comunicação entre o computador e este periférico: o computador inicia o processo, seleccionando um endereço que corresponde ao interface da impressora. Para isso usa o último conjunto de ligações (Peripheral Address Lines). Se existe um dispositivo localizado nesse endereço, ele deverá responder, assinalando ao computador a sua presença, comunicando um sinal pela linha de STATUS.
Se a resposta é positiva, o computador coloca as linhas de I/0 dirigidas como saídas; coloca em seguida os dados na linha de dados e avisa pelo sinal de STROBE que os dados estão disponíveis. Se a impressora está pronta a trabalhar, aceita os dados e imprime-os.
Uma impressora vulgar, tal como a máquina de escrever, deve seleccionar o caracter apropriado, activar o mecanismo que o vai premir contra o papel e deve movimentar-se para imprimir o próximo caracter.
Todos estes passos podem demorar cerca de 10 milisegundos (0.01 seg.).
Pode ou não ser um grande período de espera, mas o certo é que o processador tem cerca de 1 microsegundo (0.000001 seg.) para enviar o comando para a impressora. Sob o ponto de vista do computador, a impressora leva uma eternidade para o atender!
felizmente os computadores são pacientes e não remetem outro caracter, sem que a impressora esteja preparada.
A linha de FLAG conduz o sinal da impressora a pedir ao computador para esperar um pouco...

FIGURA 2

PROGRAMA POC I (contabilidade)

Existe um programa POC que funciona com cassetes e que corre no Spectrum sob vários nomes e em diversas versões, sem que saibamos quem são os autores.
Vários amigos nos têm escrito ou telefonado, solicitando que publiquemos a lista das contas do POC que nesse programa estão definidas e como se podem modificar ou substituir.
Em anexo, juntamos a listagem das contas e o MENU que aparece no écran para que possam reconhecer o programa.
O programa admite 128 contas e cerca de 1280 movimentos.
As contas estão definidas nas variáveis C(i) e C$(i), respectivamente número de conta e denominação.
Para carregar esse programa, bastará usar o clássico LOAD "" e carregado o programa, entrará imediatamente em funcionamento.
Se por qualquer razão interromper o programa, não use RUN pois isso obrigará a carregar de novo o programa, dado que destrói as variáveis com as contas definidas; em vez disso, use GOTO 50 e ficará de novo com o MENU disponível.
Se quiser, por exemplo, substituir a conta 213 — Clientes c/ LOTR por Clientes C/C e com o número 211, bastará parar o programa com STOP ou Caps Shift Break e fazer:

LET C(4) = 211 : LET C$(4) = "CLIENTES C/C"

ou, por exemplo, substituir a conta 3 (3 será o número da variável, como pode ver na listagem anexa):

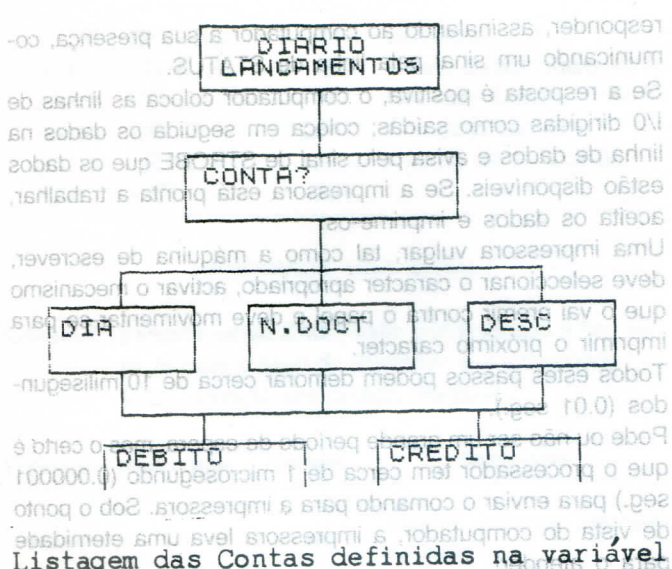
LET C(3) = 121 : LET C$(3) = "BNU"

Em seguida pode usar GOTO 50 e tudo estará de acordo com os seus desejos.
Quando terminar a operação deve gravar a nova versão do programa, com ou sem dados, com a opção 4.

```
C O N T A B I L I D A D E
        * P O C *        (POC I)

MENU :

     1 DIARIO MOVIMENTOS
     2 •   EXTRACTO
     3 FECHO MENSAL

     4      END
```

Listagem das Contas definidas na variável C(i)

```
1-11      2-12      3-211
4-213     5-216     6-219
7-221     8-223     9-229
10-23     11-241    12-242
13-243    14-244    15-245
16-248    17-25     18-261
19-262    20-263    21-264
22-268    23-269    24-274
25-279    26-28     27-29
28-311    29-312    30-317
31-318    32-32     33-33
34-34     35-35     36-36
37-38     38-39     39-41
40-421    41-422    42-423
43-424    44-425    45-426
46-429    47-43     48-44
49-47     50-48     51-49
52-52     53-55     54-56
55-57     56-58     57-59
58-611    59-612    60-617
61-618    62-62     63-6311
64-6312   65-6313   66-6314
64-6312   65-6313   66-6314
67-6315   68-6316   69-6317
70-6318   71-6321   72-6322
73-6323   74-6324   75-6325
76-6326   77-6327   78-6331
79-6333   80-6334   81-6335
82-6336   83-6337   84-6338
85-6412   86-6413   87-6414
88-6417   89-6418   90-642
91-651    92-652    93-653
94-6541   95-6542   96-657
97-658    98-661    99-662
100-663   101-664   102-668
103-67    104-68    105-69
106-711   107-712   108-713
109-717   110-718   111-72
112-73    113-75    114-76
115-77    116-78    117-79
118-81    119-821   120-822
121-823   122-824   123-8271
124-8272  125-828   126-829
127-83    128-88    129-
```

Programa POC I
Nomes das Contas definidas na variavel C$(i)

```
11    CAIXA            229   ADIANT FORNEC
12    DEP ORDEM        23    EMP CONC OBT
211   CLIENTES C/C     241   FP-I s/LUCROS
213   CLIENT C/LOTR    242   FP-I TRANSAC
216   CLIENT CB DUV    243   FP-I PROFIS
219   ADIANT CLIENT    244   FP-FUND DESEM
221   FORNECED C/C     245   FP-OUT IMP
223   FORNEC C/LOTP    248   INST PREVID
25    SOC/AC ASSOC     6323  CONSERV REP
261   CRED IMOB C/C    6324  COMUNICACAO
262   CRED IMOB C/L    6325  SEGUROS
263   REMUN PAGAR      6326  PUBLIC PROP
264   SINDICATOS       6327  TRAB ESPECIAL
268   D/C PAG DIFER    6331  TRANSP MERCAD
269   DEV CRED DIV     6333  DESLOC ESTAD
274   OUT DESP ANTE    6334  COMISS INTERM
279   OUT REC ANTEC    6335  HONORARIOS
28    PROV I s/LUCR    6336  CONTENC NOT
29    PROV CD ORE      6337  TRAB EX EXT
311   MERCADORIAS      6338  OUT SERVICOS
312   MP SUBS CONS     6412  I TRANSACCOES
317   DEV COMPRAS      6413  I SELO
318   DESC ABAT COM    6414  I s/TRANSP R
32    MERCADORIAS      6417  TAXAS
33    PROD AC SEMI     6418  OUT IMP IND
34    SUBP RES REF     642   OUT IMP DIR
35    PROD T CURSO     651   REM CORP GER
36    MP SUBS CONS     652   ORDEN SALAR
38    REGUL EXIST      653   REM ADICION
39    PROV DEP EXIS    6541  ENC CX PREV
41    IMOB FINANC      6542  ENC F DESEMP
421   TERRENOS         657   SEG AC TRAB
422   EDIFICIOS        658   OUT DESP PESS
423   EQ BAS OM INS    661   JUROS FINANC
424   FERRAM UTENS     662   DESC TITULOS
425   MAT CARGA TRA    663   DESC P PAGT
426   EQ ADM SM DIV    664   DESP S BANC
429   OUT IMOB CORP    668   OUT DESP FIN
43    IMOB INCORP      67    OUT DESP ENC
44    IMOB CURSO       68    AMORT R EXERC
47    CUSTOS PLURIE    69    PROV EXERC
48    AMORT REIN AC    711   MERCADORIAS
49    PROV IMOB FIN    712   PROD AC SEMI
52    CAPITAL SOC      713   SUB RES REF
55    RESERV LEG ES    717   DEV VENDAS
56    RESERV ESPEC     718   DESC AB VEND
57    RESERV REAVAL    72    PREST SERVIC
58    RESERV LIVRES    73    TRAB p/EMPR
59    RESULT TRANS     75    RECEIT SUPL
611   MERCADORIAS      76    REC FIN CORR
612   MP SUBS CONS     77    REC APL FIN
617   DEVOL COMPRAS    78    OUT RECEITAS
618   DESC ABAT COM    79    UTIL PROVIS
62    SUBCONTRATOS     81    RES CORR EX
6311  AGUA             821   SINISTROS
6312  ELECTRICIDADE    822   ALIEN IMOB
6313  COMBUST OUT F    823   UTIL PROV
6314  MAT CONS REP     824   REP ANUL PR
6315  FERR UT D RAP    8271  MULT FISCAIS
6316  MAT ESCRITOR     8272  MULT n/FISC
6317  MAT PUBLICID     828   OUT PERD EXT
6318  OUT FORNECIM     829   OUT GANH EXT
6321  RENDAS ALUG      83    RES EX ANTER
6322  DESP REPRES      88    RES LIQUIDOS
```

HEXA CODE

Autor: EDUARDO PARDINHA

O programa HEXA CODE e uma combinacao de de tres programas que conforme o MENU faz o seguinte. 1 Converte N. decimais em HEXA 2 Faz o PEEK da memoria para listar o seu conteudo.3 da entrada de codigo maquina na memoria, a partir do endereco calculado pelo computador, de acordo com o N. de Bytes que o programa tem.Esse N. e dado no inicio da entrada do codigo maquina.Na posicao 3 ainda temos "S" para STOP. " E " para emendar erros " L " para introduzir NOPS .A utilizacao de " E " e feita do modo seguinte:Se ocorrer um erro que so foi verificado depois de introduzido na memoria prime-se " E " seguido de ENTER faz-se esta operacao tantas vezes quantas as posicoes ja passadas depois do erro seguidamente corrige-se o erro e depois repete-se as outras posicoes ja entradas. O programa pede o que se tem de fazer.

Edurado Pardinha

```
  10 REM HEX CODE
  20 DIM A$(4)
  30 OUT 63,0: PRINT AT 5,10;"M
E N U"
  35 POKE 23609,120
  40 PRINT AT 8,5;"1-->ENTRAR No
. DECIMAL";AT 10,5;"2-->PEEK A M
EMORIA";AT 12,5;"3-->ENTRADA DE
CODIGO HEX"
  50 IF INKEY$<>"1" AND INKEY$<>
"2" AND INKEY$<>"3" THEN GO TO 5
0
  60 IF INKEY$="1" THEN GO TO 10
0
  70 IF INKEY$="2" THEN GO TO 20
0
  80 IF INKEY$="3" THEN GO TO 10
00
 100 CLS : INPUT "No.DECIMAL";B
 110 IF B=0 THEN GO TO 30
 120 GO SUB 500
 130 PRINT B$
 140 PRINT AT 21,0;"PRIMA UMA TE
CLA PARA CONT.": PAUSE 0: GO TO
100
 200 CLS: INPUT "END.INICIAL ";
X,"END.FINAL  ";Y
 210 FOR F=X TO Y: LET B=PEEK F:
 GO SUB 500
 220 PRINT B$: NEXT F: STOP
 500 FOR N=1 TO 4
 510 LET C=((B/16)-INT (B/16))*1
6
 520 LET B=INT (B/16)
 530 IF C>9 AND C<16 THEN LET A$
(N)=CHR$ (C+55)
 540 IF C<9 THEN LET A$(N)=STR$
C
 550 IF B>15 THEN NEXT N
 560 NEXT N
 570 LET B$=A$(4)+A$(3)+A$(2)+A$
(1): RETURN
1000 PRINT "Quantos Bytes tem o
programa?"
1010 INPUT FLASH 1;"Entre No.de
Bytes"; FLASH 0,a
1020 LET b=32598-a: CLEAR b
1030 LET a=PEEK 23730+256*PEEK 2
3731+1: PRINT "O programa comeca
 em: ";a
1040 PRINT "Entrar o codigo com
dois digitos"
1045 INPUT "HEx ";a$
1050 CLS : PRINT INVERSE 1;"Ende
reco";TAB 10;"Hexa";"DEC.": GO T
O 1060
1055 INPUT FLASH 1;"HEX "; FLASH
 0,a$
1060 IF a$="l" THEN GO TO 1200
1070 IF a$="e" THEN GO TO 1250
1080 IF a$="s" THEN STOP
1090 IF LEN a$<>2 THEN GO TO 105
5
1100 IF CODE a$>96 THEN LET a$(1
)=CHR$ (CODE a$(1)-32)
1110 IF CODE a$(2)>96 THEN LET a
$(2)=CHR$ (CODE a$(2)-32)
1120 IF CODE a$>70 OR CODE a$(2)
>70 THEN GO TO 1055
1130 LET b=16*CODE a$-112*(CODE
a$>57)+CODE a$(2)-7*(CODE a$(2)>
57)-816
1140 POKE a,b
1150 PRINT a;TAB 10;a$,b
1160 LET a=a+1
1170 GO TO 1055
1200 LET b=PEEK a
1210 LET c=INT (b/16)
1220 LET a$=CHR$ (c+48+7*(c>9))+
CHR$ (b-16*c+48+7*((b-16*c)>9))
1230 GO TO 1150
1250 LET a=a-1
1260 POKE 23689,PEEK 23689+1
1270 GO TO 1055
```

Por vezes surge a necessidade de converter numeros HEXa em decimal ou decimalem qualquer base. E claro que existem ja tabelas elaboradas para esse fim, mas eu lembrei-de fazer dois pequenos programas que satisfazem essas condicoes.
Estes programas depois de passados para o computador e de fazer RUN pede o que nos convem converter.

22/12/1984

Eduardo Pardinha

```
   5 REM *Conversao de qualquer
numero em qualquer base*
  10 PRINT "Conversao de N.em qu
alquer base"
  15 LET A$=""
  20 INPUT "N.a converter ?  ";A
  30 PRINT '"N.Decimal = ";A
  35 INPUT "Base pertendida ?  "
;B
  50 LET C=A
  60 IF B<>16 THEN GO SUB 500
  70 IF B=16 THEN DIM A$(4): GO
SUB 520
 100 LET A=INT (A/B)
 110 IF A>0 THEN GO TO 55
 120 PRINT '''C;" Na Base ";B ;
"  e = ";A$
 130 PAUSE 0: RUN
 500 LET A$=CHR$ (48+A-INT (A/B)
*B)+A$
 510 RETURN
 520 FOR n=1 TO 4
 530 LET C=((A/B)-INT (A/B))*B
 540 LET A=INT (A/B)
```

```
550 IF C>9 AND C<16 THEN LET A$
(n)=CHR$ (C+55)
560 IF C<10 THEN LET A$(n)=STR$
C
570 IF A>15 THEN NEXT n
580 NEXT n
590 LET A$=A$(4)+A$(3)+A$(2)+A$
(1): RETURN
  1 POKE 23658,8
  5 REM *CONVERSAO DE N.HEXA EM
DECIMAL*
 10 INPUT "N. EM HEXADECIMAL
";B$
 20 IF B$="" THEN STOP
 30 FOR V=1 TO LEN B$
 40 IF CODE B$(V)<47 OR CODE B$
(V)>70 THEN GO TO 10
 50 NEXT V
 60 FOR F=1 TO 4-LEN B$: LET B$
="0"+B$
 70 NEXT F
 80 LET C=0
 90 LET X=4
100 FOR F=0 TO 3
110 IF CODE B$(X)>64 AND CODE B
$(X)<71 THEN LET A=CODE B$(X)-55
120 IF CODE B$(X)>47 AND CODE B
$(X)<58 THEN LET A=CODE B$(X)-48
130 LET X=X-1
140 LET B=16↑F*A: LET C=C+B
150 NEXT F
160 PRINT B$,C
170 GO TO 10
180 SAVE "HEX.DEC." LINE 1
```

CONSTELAÇÕES

Constelações e um pequeno programa educacional para mostrar as figuras mais conhecidas das constelações do Hemisferio Norte.
Pode (e deve) ser modificado e ampliado.
Se quizer maior realismo, introduza a linha 1

1 BORDER 0 : PAPER 0 : INK 7

```
                                      *
 URSA MAIOR


                *   *     *           *


            *              *
```

```
                             * estr.Polar
 URSA MENOR             *

                      *


                *  *


               *  *
```

```
seleccionar uma constelacao

1=URSA MAIOR
2=CASSIOPEIA
3=URSA MENOR
4=CEPHEUS
5=BOIEIRO
6=LEAO
```

```
  10 PRINT "seleccionar uma cons
telacao"
  20 PRINT
  30 PRINT "1=URSA MAIOR"
  31 PRINT "2=CASSIOPEIA"
  32 PRINT "3=URSA MENOR"
  33 PRINT "4=CEPHEUS"
  34 PRINT "5=BOIEIRO"
  35 PRINT "6=LEAO"
  40 INPUT A
  45 PAUSE 150
  46 CLS
  50 IF A=1 THEN GO TO 60
  51 IF A=2 THEN GO TO 100
  52 IF A=3 THEN GO TO 140
  53 IF A=4 THEN GO TO 180
  54 IF A=5 THEN GO TO 220
  55 IF A=6 THEN GO TO 260
  60 GO SUB 1000
  76 PRINT AT 13,6;"*";AT 10,10;
"*";AT 10,13;"*";AT 10,18;"*";AT
13,21;"*";AT 10,28;"*";AT 6,28;
"*"
  80 PRINT TAB 1;"URSA MAIOR"
  90 GO TO 2000
 100 GO SUB 1000
 110 PRINT AT 4,10;"*";AT 8,4;"*
";AT 9,12;"*";AT 10,15;"*";AT 12
,16;"*"
 120 PRINT TAB 1;"CASSIOPEIA"
 130 GO TO 2000
 140 GO SUB 1000
 150 PRINT AT 15,9;"*";AT 15,12;
"*";AT 11,12;"*";AT 11,10;"*";AT
7,15;"*";AT 5,17;"*";AT 4,20;"*
";" estr.Polar"
 160 PRINT TAB 1;"URSA MENOR"
 170 GO TO 2000
 180 GO SUB 1000
 190 PRINT AT 8,2;"*";AT 10,6;"*
";AT 11,12;"*"
 200 PRINT TAB 1;"CEPHEUS"
 210 GO TO 2000
 220 GO SUB 1000
 230 PRINT AT 3,8;"*";AT 4,4;"*"
;AT 5,11;"*";AT 8,5;"*";AT 8,8;"
*";AT 13,5;"*"
 240 PRINT AT 19,1;"BOIEIRO"
 250 GO TO 2000
 270 PRINT AT 1,13;"*";AT 1,15;"
*";AT 3,13;"*";AT 5,14;"*";AT 5,
16;"*";AT 7,7;"*";AT 8,18;"*";AT
 9,8;"*";AT 10,4;"*"
 280 PRINT TAB 1;"LEAO"
 290 GO TO 2000
1000 FOR N=1 TO 22
1010 PRINT "
"
1020 NEXT N
1030 RETURN
2000 PAUSE 0
2010 RUN
```

NÚMEROS MÁGICOS

SPECTRUM

Depois de passada a listagem, grave o programa e verifique a gravação.
Por ex.: SAVE "NUM. MÁG." LINE 1 e VERIFY "".
Agora faça RUN. Escolha um número entre 60 e 4290. Segue-se o desenho de um quadro de 5/5 com números ao «acaso», dos quais terá de escolher 5, um por cada vez (accione a tecla com a letra correspondente ao n.º). Repare que a coluna e a linha, com início no número escolhido, serão apagadas. Finalmente, os cinco números são somados e o resultado obtido é, concerteza, o número que inicialmente registou.
Divirta-se!

```
   1 PRINT "NUMEROS MAGICOS": PA
USE 50: BORDER 0: PAPER 0: INK 7
: CLS
   2 POKE 23609,50
 100 INPUT "Limite (60-4920) ";f
 110 IF f<>INT f OR f<60 OR f>49
20 THEN GO TO 100
 120 BEEP .1,16: PRINT AT 7,27;"
=";AT 7,(28+(2 AND f<100)+(1 AND
 f>99 AND f<1000)); BRIGHT 1;f
 130 DIM a(5,5)
 140 LET k=(f-60)/5
 150 FOR z=1 TO 5
 160 FOR y=1 TO 5
 170 LET a(z,y)=INT k+5*(z-1)+y-
1+(5*(k-INT k) AND y=5)
 180 PRINT AT 3*z,5*y+1; INK 5;C
HR$ (96+5*(z-1)+y)
 190 PRINT AT 3*z-1,5*y-2; BRIGH
T 1; INK 7;a(z,y): BEEP .07,5*z+
y-20
 200 NEXT y
 210 NEXT z
 220 FOR z=0 TO 5
 230 PLOT 16,164-24*z: BEEP .06,
3*z
 240 DRAW 200,0
 250 PLOT 16+40*z,164: BEEP .03,
3*z+2
 260 DRAW 0,-120
 270 NEXT z
 280 FOR u=32 TO 48: BEEP .05,u:
 NEXT u
 290 PRINT AT 18,0;"escolhe um n
umero."
 300 DIM g(5,2)
 310 FOR z=1 TO 4
 320 LET i=CODE INKEY$-96
 330 IF i<1 OR i>25 THEN GO TO 3
20
 350 LET r=INT (i/5+.9)
 360 LET c=i-5*(r-1)
 370 IF g(r,1)<>0 THEN BEEP 1,16
: GO TO 320
 380 IF g(c,2)<>0 THEN BEEP 1,16
: GO TO 320
 390 BEEP .2,i
 400 LET g(r,1)=1
 410 LET g(c,2)=1
 420 FOR q=1 TO 5
 430 PRINT AT 3*r-1,5*q-2; INK 2
;a(r,q)
 440 PRINT AT 3*q-1,5*c-2; INK 2
;a(q,c)
 450 BEEP .01,q: BEEP .02,6*q
 460 NEXT q
 470 PRINT AT z,28;"+";AT 1,28;"
";AT z,(29+(2 AND a(r,c)<10)+(1
 AND a(r,c)>9 AND a(r,c)<100));
BRIGHT 1;a(r,c)
 480 NEXT z
 490 PRINT AT 18,0;"Transferindo
 o ultimo numero..."
 500 FOR z=1 TO 5
 510 BEEP .5,10*z
 520 IF g(z,1)=0 THEN LET r=z
 530 IF g(z,2)=0 THEN LET c=z
 540 NEXT z
 550 PRINT AT 5,28;"+";AT 5,(29+
(2 AND a(r,c)<10)+(1 AND a(r,c)>
9 AND a(r,c)<100)); BRIGHT 1;a(r
,c)
 560 PRINT AT 3*r-1,5*c-2; INK 2
;a(r,c)
 570 PRINT AT 18,0; INK 6;"Agora
, soma os numeros que       escol
heste. O total e...";f;" !"'" O
 TEU NUMERO! "
 580 FOR n=54 TO 64: BEEP .01,RN
D*30: OUT 254,n/8: NEXT n
 590 PRINT #0;"Tecla 1 para reco
mecares"
 600 PAUSE 0
 610 IF INKEY$="1" THEN RUN 100
 620 STOP
```

88 a	89 b	90 c	91 d	92 e	
93 f	94 g	95 h	96 i	97 j	= 500
98 k	99 l	100 m	101 n	102 o	
103 p	104 q	105 r	106 s	107 t	
108 u	109 v	110 w	111 x	112 y	

escolhe um numero.

TOP 10 EM INGLATERRA

1 — LORDS OF MIDNIGHT
2 — SABRE WULF
3 — FULL THROTTLE
4 — GO TO JAIL
5 — DALEY'S DECATHLON
6 — JUMP CHALLENGE
7 — AVALON
8 — PYJAMARAMA
9 — MUGSY
10 — COMBAT LYNX

NO CLUB Z 80

(Os mais vendidos)

1 — DEUX EX MACHINA
2 — SHERLOCK HOLMES
3 — BEACH HEAD
4 — CHUCKIE EGG
5 — MACHINE CODE TUTOR
6 — POLE POSITION
7 — DANGER MOUSE
8 — PHEENIX
9 — JUMP CHALLENGE
10 — FULL THROTTLE

PUZZLE

Traduzido

Adaptado: CLUBE Z80

Traduzido e adaptado pelo Clube Z80

Experimente encontrar no quadrado as palavras que vão sendo assinaladas pelo Spectrum.

Apenas verifique a sua posição (diagonal; horizontal nos dois sentidos; vertical)... não deve dar qualquer resposta ao computador.

Veja se consegue obter a resposta em primeiro lugar.

Introduza o programa tal como nas listagens e grave-o utilizando a linha 9000.

GOTO 9000 e depois verifique a gravação.

```
   1 DATA "ZX"
   2 DATA "WL"
   3 DATA "RAT"
   4 DATA "BIT"
   5 DATA "ROM"
   6 DATA "RAM"
   7 DATA "BUS"
   8 DATA "TAPE"
   9 DATA "PORT"
  10 DATA "BYTE"
  11 DATA "CHIP"
  12 DATA "BAUD"
  13 DATA "CP/M"
  14 DATA "DRIVE"
  15 DATA "BASIC"
  16 DATA "EPROM"
  17 DATA "CLIVE"
  18 DATA "SYNTAX"
  19 DATA "BUFFER"
  20 DATA "MEMORY"
  21 DATA "CHEETAH"
  22 DATA "PROGRAM"
  23 DATA "MONITOR"
  24 DATA "MACHINE"
  25 DATA "SOFTWARE"
  26 DATA "SPECTRUM"
  27 DATA "JOYSTICK"
  28 DATA "COMPUTER"
  29 DATA "CASSETTE"
  30 DATA "CONQUEST"
  31 DATA "SINCLAIR"
  32 DATA "HARDWARE"
  33 DATA "BATATTACK"
  34 DATA "INTERFACE"
  35 DATA "CARTRIDGE"
  36 DATA "DRAGONFIRE"
  37 DATA "BEARGEORGE"
  38 DATA "MOONSWEEPER"
  39 DATA "SWEETTALKER"
  61 DATA "BUSCOYCOMPUTERW"
  62 DATA "AERATYRUHATEEHC"
  63 DATA "UIOSOPDOASDPFGH"
  64 DATA "DHJSMAROMJEVIRD"
  65 DATA "EJREKLATTEEWSKL"
  66 DATA "GLZTQXGCWVMTBNM"
  67 DATA "RATTMQOSPECTRUM"
  68 DATA "OKQEWENYIECRTOT"
  69 DATA "ECYUIOFNONPIASP"
  70 DATA "GIDMONITORCXSFH"
  71 DATA "RTHMJKRALZCLZAV"
  72 DATA "ASEXBNEXPIHCATB"
  73 DATA "EYNPMECAFRETNIO"
  74 DATA "BOERAWTFOSETYBR"
  75 DATA "RJTTATEERAWDRAH"
 100 LET dif=0: LET total=0
 110 CLS : RESTORE : REM escreve
 palavras
 120 FOR n=1 TO 39
 130 READ n$
 140 PRINT AT 15+n-6*(n>6)-6*(n>
12)-21*(n>18),5*(n>6)+6*(n>12)+8
*(n>18);n$
 150 NEXT n
 160 DIM a$(15,15): REM quadrado
 170 FOR a=1 TO 15: READ a$(a)
 180 PRINT PAPER 6;AT a,1;a$(a)
 190 NEXT a
 199 REM procura
 200 FOR n=1 TO 39: LET found=0
 210 RESTORE n: READ n$: PRINT F
LASH 1;AT 15+n-6*(n>6)-6*(n>12)-
21*(n>18),5*(n>6)+6*(n>12)+8*(n>
18);n$
 220 LET len=LEN n$: LET l=len-1
 230 FOR b=1 TO 15: FOR a=1 TO 1
5
 240 IF n$(1)<>a$(b,a) THEN GO T
O 300
 250 FOR h=-1 TO 1: REM direccao
 horizontal
 255 IF h=-1 AND a-l<1 THEN GO T
O 280
 256 IF h=1 AND a+l>15 THEN GO T
O 290
 260 FOR v=-1 TO 1: REM direccao
 vertical
 264 IF v=-1 AND b-l<1 THEN GO T
O 280
 265 IF h=0 AND v=0 THEN GO TO 2
80
 266 IF v=1 AND b+l>15 THEN GO T
O 280
 270 GO SUB 500: REM procura
 280 NEXT v
 290 NEXT h
 300 NEXT a: NEXT b
 310 LET dif=dif+found: PRINT BR
IGHT found;AT 15+n-6*(n>6)-6*(n>
12)-21*(n>18),5*(n>6)+6*(n>12)+8
*(n>18);n$
 320 NEXT n
 330 PRINT #0;AT 0,0;dif;" palav
ras encontradas.";total;" ao tod
o."
 340 PAUSE 4e4: STOP
 500 REM sobrotina procura
 550 FOR t=1 TO l
 560 LET x=a+h*t: LET y=b+v*t
 570 IF n$(1+t)<>a$(y,x) THEN RE
TURN
 580 NEXT t
 590 LET total=total+1: LET foun
d=1: REM encontradas
 600 PRINT FLASH 1; BRIGHT 1;AT
15+n-6*(n>6)-6*(n>12)-21*(n>18),
5*(n>6)+6*(n>12)+8*(n>18);n$
 610 FOR q=1 TO 5
 620 FOR t=0 TO l
 630 LET x=a+h*t: LET y=b+v*t
 640 PRINT BRIGHT 0;AT y,x;n$(1+
t)
 650 BEEP .01,x-2*y
 660 PRINT BRIGHT 1;AT y,x;n$(1+
t)
 670 NEXT t
 680 NEXT q
 700 RETURN
9000 SAVE "palavras" LINE 9020
9010 VERIFY "palavras"
9020 CLS : PRINT AT 6,5;"PUZZLE
DE PALAVRAS"
9030 PAUSE 200: GO TO 1
```

QUEM PODE AJUDAR?

Problemas colocados por VICTOR MAIA/Porto

- Porque será que no jogo «4 EM LINHA» publicado no n.º 25, não entra o movimento:

0	180	2
0	290	1

- No programa «PUZZLE DE PALAVRAS» a instrução «um momento» persiste indefinidamente?

ANIMAÇÃO

Aqui vai um pequeno programa que demonstra como se pode fazer animação nos programas.

Ele produz um pequeno desenho de um boneco atravessando o ecran da direita para a esquerda.

O movimento é do tipo desenho animado.

ATENÇÃO: As linhas 110-120-130 — só devem dar entrada no modo gráfico após o RUN inicial. Antes disso, deve introduzi-las sob a forma.

110 LET a$ = "acbd"
120 LET b$ = "egfh"
130 LET c$ = "ikjl"

Depois de ter introduzido todo o programa deve fazer RUN. Quando aparecerem letras no ecran, deve fazer BREAK e chamar de novo essas linhas. Escreva-as de novo, mas agora deve escrever a, b, c, d, e, f, g, h, i, j, k, l, em modo GRÁFICO ou seja com o cursor em **G.**

```
  10 FOR f=0 TO 11
  20 FOR n=0 TO 7
  30 READ a: POKE USR CHR$ (144+
f)+n,a
  40 NEXT n: NEXT f
  60 DATA 0,0,0,0,0,0,0,0,2,1,0,
0,0,0,0,3,0,0,0,0,0,112,113,32,1
20,164,36,96,80,208,156,132
  80 DATA 0,0,0,0,1,1,0,0,0,0,3,
0,0,0,0,3,0,0,0,0,192,192,128,19
2,224,208,224,192,192,192,192,19
2
 100 DATA 0,0,0,0,0,28,92,72,62,
9,9,56,68,66,193,7,0,0,0,0,0,0,0
,0,0,0,0,0,0,0,0,0
 110 LET a$="    "
 120 LET b$="    "
 130 LET c$="    "
 140 FOR a=30 TO 0 STEP -1
 150 FOR f=0 TO 2
 155 READ q$
 160 PRINT AT 10,a;q$( TO 2);AT
11,a;q$(3 TO )
 170 PAUSE 8
 180 NEXT f
 190 RESTORE 155
 195 NEXT a
 200 DATA a$,b$,c$
 210 CLS : GO TO 140
```

MODO MUSICAL

SPECTRUM

Não se trata propriamente de um programa de música, mas de uma indicação sonora do MODO ("MODE"), em que está a trabalhar o Spectrum num determinado momento ("E", "C", "K", "G" ou "L").

O PROGRAMA:

Introduza a primeira linha (primeira parte) e grave-a.

Ex.: SAVE "MÚSICA" LINE 1

Pode retirar agora esta linha e iniciar a segunda parte com a listagem do carregador hex.

Depois de introduzido, faça RUN e dê entrada do código-máquina da listagem 3.

(Oito bytes por cada vez, ou seja, duas linhas e sem espaços entre si).

No final de cada duas linhas ser-lhe-á pedida a verificação, para a qual deve dar o valor correspondente a essa entrada.

(Ex.: Primeira linha é 441)

```
   1 CLEAR 64739: LOAD ""CODE
 RANDOMIZE USR 65105

   5 POKE 23658,8
   9 CLEAR 60000
  10 FOR i=64740 TO 65140 STEP 8
  20 LET cs=0
  30 PRINT AT 0,0;"Endereco ";i
  40 INPUT "8 bytes ", LINE a$
  60 IF LEN a$<>16 THEN GO TO 10
00
  80 LET f=0: FOR j=1 TO 16
  90 IF (a$(j)<"0" OR a$(j)>"9")
 AND (a$(j)<"A" OR a$(j)>"F") TH
EN LET f=1
 100 NEXT j
 102 IF f=1 THEN GO TO 1000
 105 FOR n=0 TO 7
 120 LET y=CODE a$(1)-48: IF y>9
 THEN LET y=y-7
 130 LET z=CODE a$(2)-48: IF z>9
 THEN LET z=z-7
 140 LET va=16*y+z
 150 LET cs=cs+va
 160 POKE i+n,va
 165 PRINT AT 2,n*3;a$( TO 2)
 170 LET a$=a$(3 TO )
 180 NEXT n
 183 INPUT "Verificacao "; LINE
a$
 184 PRINT AT 2,25;a$
 185 IF VAL a$<>cs THEN GO TO 10
00
 187 CLS
 190 NEXT i
 200 CLS : PRINT "Grave o BASIC
antes do cod.maq." "Retire o ca
bo de EAR"
 210 SAVE "IM 2 BEEP"CODE 64740,
408
 220 CLS : PRINT "VERFICAR"
 230 VERIFY ""CODE
 240 CLS : PRINT "OK!": STOP
1000 PRINT AT 15,0;"ERRO": GO TO
 20
```

```
64740    21 08 5C 7E
64744    36 00 21 5F    441
64748    FE 77 AF 7E
64752    FE 00 CA 50    1210
64756    FE 7E FE 0D
64760    CA 36 FD AF    1331
64764    7E FE E5 D2
64768    8C FD AF 7E    1514
64772    FE 80 D2 0C
64776    FD C3 13 FD    1324
64780    AF 7E FE A4
64784    DA BE FD AF    1555
64788    7E FE 61 D2
64792    1D FD C3 24    1200
64796    FD AF 7E FE
64800    7B DA DE FD    1624
64804    AF 7E FE 41
64808    30 02 18 07    701
64812    AF 7E FE 5B
64816    DA FD FD C3    1565
```

```
64820  25 FE F5 11
64824  10 00 21 C8   802
64828  00 CD B5 03
64832  11 10 00 21   455
64836  98 00 CD B5
64840  03 11 10 00   574
64844  21 C8 00 CD
64848  B5 03 11 10   655
64852  00 21 98 00
64856  CD B5 03 11   591
64860  10 00 21 C8
64864  00 CD B5 03   638
64868  11 10 00 21
64872  98 00 CD B5   604
64876  03 11 10 00
64880  21 C8 00 CD   474
64884  B5 03 11 10
64888  00 21 98 00   402
64892  CD B5 03 11
64896  10 00 21 C8   655

64900  00 CD B5 03
64904  F1 C3 4B FE   1154
64908  F5 11 10 00
64912  21 08 01 CD   525
64916  B5 03 11 10
64920  00 21 F8 00   498
64924  CD B5 03 11
64928  10 00 21 E8   687
64932  00 CD B5 03
64936  11 10 00 21   455
64940  D8 00 CD B5
64944  03 11 10 00   638
64948  21 C8 00 CD
64952  B5 03 F1 C3   1058
64956  4B FE F5 11
64960  10 00 21 FC   892
64964  01 CD B5 03
64968  11 10 00 21   456
64972  28 01 CD B5
64976  03 11 10 00   463

64980  21 FC 01 CD
64984  B5 03 F1 C3   1111
64988  4B FE F5 11
64992  10 00 21 C8   840
64996  00 CD B5 03
65000  11 10 00 21   455
65004  C6 05 CD B5
65008  03 11 10 00   625
65012  21 28 02 CD
65016  B5 03 F1 18   729
65020  4E F5 11 10
65024  00 21 B4 01   570
65028  CD B5 03 11
65032  10 00 21 C8   655
65036  00 CD B5 03
65040  11 10 00 21   455
65044  D0 0A CD B5
65048  03 11 10 00   640
65052  21 F8 00 CD
65056  B5 03 F1 18   935

65060  26 F5 11 10
65064  00 21 C8 02   551
65068  CD B5 03 11
65072  10 00 21 64   555
65076  01 CD B5 03
65080  11 10 00 21   456
65084  C8 00 CD B5
65088  03 11 10 00   622
65092  21 78 00 CD
65096  B5 03 F1 21   816
65100  08 5C 36 00
65104  C9 21 08 5C   488
65108  36 00 3E 09
65112  ED 47 F3 ED   913
65116  5E FB C9 00
65120  3E 3F ED 47   979
65124  F3 ED 56 FB
65128  C9 F3 F5 C5   1703
65132  D5 E5 DD E5
65136  CD E4 FC DD   1798

65140  E1 E1 D1 C1
65144  F1 C3 38 00   1344
```

ON ERROR GOTO

SPECTRUM/48 K

Traduzido

Adaptado: CLUBE Z80

Para quem desconhece este comando...

Computadores tais como BBC, Apple e outros de preços menos acessíveis, podem receber esta instrução:

ON ERROR GOTO 1000

Isto significa que em caso de ocorrência de erro como: "Integer ou of range"; "Out of memory"... o computador executaria o programa a partir da linha 1000, não transmitindo tais mensagens. Esta linha (1000) pode conter mensagens que expliquem ao utilizador o motivo e o tipo de erro, evitando-se assim o BREAK do programa.

A rotina que a seguir é apresentada tem aplicação no Spectrum e vai permitir a operação que acima indicamos.

A linha para onde o programa saltará em caso de erro, deve ser indicada pela variável "ERROR", sendo esse número de linha o valor que lhe for atribuído. Se a variável não existe, por não ter sido definida, ou se está igual a 0, não haverá salto no programa, logo terá as mensagens habituais. Terá a mensagem "OK" se atribuir a "ERROR" um valor a que não corresponda a um número de linha existente no programa. Ex.: GOTO 9999. Se não existe esta linha no seu programa, não pode atribuir a "ERROR" o valor 9999.

A rotina cria também 2 novas variáveis, ou altera-as, se já existiam, chamadas "ERR" e "ERL".

A variável "ERR", corresponde aos códigos de erro 0 – 27 (0 = OK; 4 = Out of memory, etc.).

Veja o seu manual.

A variável "ERL" corresponde ao n.º de linha que ocasionou o erro.

Não haverá Salto nas mensagens seguintes:

STOP statement.
BREAK — CONT repeats
STOP no programa
BREAK no programa
OK

Para estes casos, pode dar entrada de:

FOR F = 60125 TO 60144 : POKE F,0 : NEXT F
(Spectrum 48 K)

Assim apenas pode ocorrer a mensagem "OK"; mas, atenção, deve ter a certeza que a linha para a qual pretende o salto não contenha erro também erro, pois nesse caso o programa saltará sempre para essa posição não permitindo o "BREAK".

```
 170 CLEAR 59999
 180 PRINT OVER 1;AT 8,13;"ESPER
E";AT 10,14;"POR FAVOR"
 190 PLOT 129,30: DRAW OVER 1;0,
145,253*PI
 200 DATA "33882AB25C363E2BF92B2
B223D5C1807FD363102CD9517CDB0163
E00CD0116FD3600FFCD2C0FCD171BFDC
```

```
B007E2013FDCB3066CAD5EA2A595CCDA
711FD3600FF18D82A595C225D5CCDFB1
978B1C208ECDFFE0DCA6EEAFDCB3046C
4AF0DCD6E0D3E19FD964F328C5CFDCB0
1FEFD3600FFFD"
 210 DATA "28E9360A01CD8A1B3A3A5
CFEFFCA7AEBFE08CA7AEBFE0CCA7AEBF
E10CA7AEBFE14CA7AEB2168EC225D5CC
DB22830132A595C2B010800CD5516233
6A52336722336F2233A3A5C3C3600233
6002377233600233600216CEC225D5CC
DB22830132A595C2B010800CD5516233
6A52336722336"
 220 DATA "2CEFEC23ED4B455C36002
336002371237023360021 62EC225D5CC
DB228381F23CDB433CDA22D78B128143
E0032445C3D323A5CED43425C11D5EAD
5C3761B76FDCB01AEFDCB304EC4DF0E3
A3A5C3CF5210000FD7437FD742622 0B5
C21010022165CCDB016FDCB37AECD6E0
DFDCB02EEF147"
 230 DATA "2B1FFE0A3802C607CDEF1
53E20D7781191 13CD0A0CAF113615CD0
A0CED4B455CCD1B1A3E3AD7FD4E0D060
0CD1B1ACD97103A3A5C3C281BFE09280
4FE152003FD340D0103001170 5C21445
CCB7E280109EDB8FD360AFFFDCB019EC
375EAED43495C2A5D5CEB215AECE52A6
15C37ED52E560"
 240 DATA "20BF69CD6E192006CDB81
9CDE819C1793DB02828C503030303 2BE
D5B535CD5CD5516E122535CC1C5132A6
15C2B2BEDB82A495CEBC1702B712B732
B72F1C36EEA3E0A010000C38AEB45525
24F520D4552520D45524C0D000000000
00000000000000000000000000000000
0000000000000"
1000 LET A=10: LET B=11: LET C=1
2: LET D=13: LET E=14: LET F=15:
 LET H=60000: LET I=200
1010 FOR G=1 TO 5
1020 READ A$: LET S=VAL A$(1)*40
96+VAL A$(2)*256+VAL A$(3)*16+VA
L A$(4): LET A$=A$(5 TO )
1030 LET T=VAL A$(1)*16+VAL A$(2
): LET S=S-T: POKE H,T: LET H=H+
1: LET A$=A$(3 TO ): IF A$<>"" T
HEN GO TO 1030
1040 IF S<>0 THEN PRINT AT 19,0;
"ERRO NA LINHA ";I: STOP
1050 LET I=I+10
1060 NEXT G
1070 PRINT OVER 1;AT 8,13;"ESPER
E";AT 10,12;"POR FAVOR"
1080 PRINT OVER 1;AT 9,15;"OK!"
1090 PLOT OVER 1;129,30: DRAW OV
ER 1;0,145,253*PI
1100 LET A$="USE:  RANDOMIZE USR
 60000  "
1110 FOR F=1 TO LEN A$: PRINT AT
 19,32-F;A$( TO F): BEEP .02,20:
 PAUSE 4: NEXT F
```

O SPECTRUM E O USO COMANDO DRAW

Quando tentamos produzir gráficos no Spectrum temos de dar continuamente os valores de cada ponto ou o deslocamento das coordenadas.

A rotina que publicamos em seguida permite poupar tempo e evitar esse cálculo.

Após o programa 1 ter sido verificado, guarde-o em cassete usando o procedimento SAVE «D» CODE 32000,50.

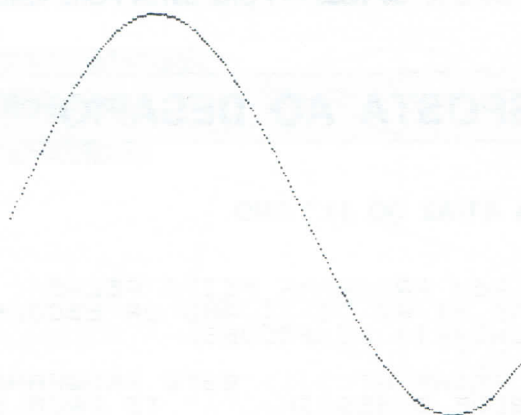
Quando necessitar de usar DRAW e PLOT num dos seus programas, carregue esse pequeno programa, e use-o chamando com RANDOMIZE USR 32000, o que fará traçar uma linha desde o último ponto escrito no ecran até ao novo ponto especificado pelos seguintes valores POKE 32007 — coordenada X.
POKE 32027 — coordenada Y.

Para inicializar o curso gráfico, pode alterar o valor da variável do sistema.

COORDENADAS: POKE 23677, X
POKE 23678, Y

A segunda listagem trata apenas de um exemplo de utilização da rotina.

```
PROGRAMA 1...correspondente ao
codigo maquina para calcular os
pontos de PLOT e DRAW


  10 FOR n=32000 TO 32049
  20 READ p0
  30 POKE n,p0
  40 NEXT n
  50 DATA 217,229,217,33,125,92,
62,250,150,79,30,1,210,23,125,30
,255,126,33,7,125,150,79,33,126,
92,62,69,150,71,22,1,210,43,125,
22,255,126,33,27,125,150,71,205,
186,36,217,225,217,201
```

Programa exemplo do uso da rotina de DRAW

```
   3 LOAD ""CODE
   5 LET step=1
   6 LET graph=1000
   7 LET a$="usar plot..."
  10 FOR f=1 TO 2
  15 PAUSE 30: CLS
  20 POKE 23677,0
  30 POKE 23678,80
  40 PRINT a$
  50 FOR x=0 TO 255 STEP step
  60 LET y=80+75*SIN (x/128*PI)
  70 GO SUB graph
  80 NEXT x
  90 LET step=10
  95 LET graph=500
  96 LET a$="usar draw TO (ate..
.)"
  98 NEXT f
 101 STOP
 500 POKE 32007,x
 510 POKE 32027,y
 520 RANDOMIZE USR 32000
 530 RETURN
1000 PLOT x,y
1001 RETURN
```

FACILIDADES

JOGOS/SPECTRUM

Traduzido
Adaptado: CLUBE Z80

Para aumentar o n.º de vidas, diminuir o grau de dificuldade, alterar o esquema de jogo, aqui vão os POKES para alguns dos já bem conhecidos jogos da ULTIMATE.

JETPAC:
— Para aumentar o n.º de vidas — POKE 25373,255 POKE 26075,0 — permite o lançamento do ROCKET apenas com 1 das 6 medidas de combustível.

PSSST:
— Aumentar n.º de vidas — POKE 24984,0

ZZOOM:
— Aumentar n.º de vidas — POKE 24743,0

ATICATAC:
— Aumentar n.º de vidas — POKE 36519,0

LUNAR JETMAN:
— POKE 36965,0 e repare que será mais fácil continuar.

OUTROS...

MOON ALERT:
— Aumentar o n.º de vidas — POKE 39754 POKE 42404, x (x — até 14 vidas, máximo) e POKE 42654,195 para a imortalidade.
POKE 42249,24 para parar a contagem.

PSYTRON:
— Para aumentar o combustível — (carregar o jogo, excepto a última parte) — fazer: BORDER 7, e POKE 28625,0: POKE 28626,0; para aumentar n.º de vidas — POKE 41098,17: POKE 41099,32: POKE 411001: POKE 41101,0 e para aumentar oxigénio: POKE 26142,62: POKE 26143,255: POKE 26144,0.

FRANK N STEIN:
— POKE 28277, x (x, vidas).

Bom, mas o que fazer com tudo isto?

Primeiro carregue o jogo usando MERGE"" (não para todos os casos), tal como quando usa LOAD"" para carregar o jogo. Quando aparecer a mensagem «OK», pare o gravador. Faça LIST e agora altere com as várias sugestões. Assim que terminado: RUN e <ENTER> e re-inicie o gravador.

RESPOSTA AO DESAFIO

TURMA A1/A2 DO 11.º ANO

```
   0>REM PROGRAMA FEITO PELAS
TURMAS A1/A2 DO 11 ANO DA ESCOLA
 SECUNDARIA DE SILVES
  10 CLS
  20 PRINT AT 1,1;"ESTE PROGRAMA
 RESOLVE O SEGUIN-"'''" TE PROBLE
MA:"''''"  -CALCULAR A AREA DUM R
ECTANGU-"''"  LO INSCRITO NUMA
CIRCUNFEREN-"''"  CIA,CONHECIDA
S AS MEDIDAS DO"''"  SEU RAIO E
 DE UMA DAS DIMEN-"''"  SOES DO
 RECTANGULO."
  30 PRINT AT 21,8;"PRIMA UMA TE
CLA": PAUSE 0: PRINT AT 21,8;"
               "
  40 INPUT "RAIO DO CIRCULO: r="
;r
  50 IF r<=0 THEN PRINT AT 21,0;
"TEM QUE SER r>0": GO TO 40
  60 INPUT "MEDIDA DUM LADO DO R
ECTANGULO:     a=";a
  70 IF a<=0 THEN PRINT AT 21,0;
"TEM QUE SER a>0": GO TO 60
  80 IF a>=2*r THEN PRINT AT 20,
0;"SO PODE INSCREVER-SE NO CIRCU
LO DESDE QUE a<2*r": GO TO 60
  90 BRIGHT 1: CLS
 100 CIRCLE 63,113,57
 110 PLOT 13,89: DRAW 101,0: PRI
NT AT 11,5;"a=";a
 120 PRINT AT 16,0;"ESTE E' UM D
OS LADOS DO RECTAN- GULO DE QUE
SE PRETENDE CALCULARA AREA."
 130 GO SUB 480
 140 PRINT AT 16,0;"O RECTANGULO
,DE LADOS a e b ,E'"
 150 PLOT 114,89: DRAW 0,48: DRA
W -101,0: DRAW 0,-48
 160 FOR x=13 TO 114 STEP 2: FOR
 y=89 TO 137 STEP 2: PLOT x,y: N
EXT y: NEXT x
 170 GO SUB 480
 180 PRINT AT 16,0;"VAMOS DESLOC
AR O RECTANGULO"
 190 FOR x=13 TO 114 STEP 2: FOR
 y=89 TO 137 STEP 2: PLOT INVERS
E 1;x,y: PLOT x+131,y+32: PAUSE
2: NEXT y: NEXT x
 200 PLOT 62,113: PLOT 63,113: P
LOT 64,113: PLOT 63,112: PLOT 63
,114
 210 GO SUB 480
 220 PRINT AT 16,0;"COMO O DIAME
TRO,2*r,DO CIRCULO  E' UMA DIAGO
NAL DO RECTANGULO"
 230 PLOT 13,89: DRAW 101,48
 240 PRINT AT 7,5;"2r";AT 7,13;"
b"
 250 GO SUB 480
 260 PRINT AT 16,0;"FORMA-SE UM
TRIANGULO RECTANGULO"
 270 FOR y=0 TO 48: FOR x=13+2*y
+4 TO 114 STEP 3: PLOT x,y+89: N
EXT x: NEXT y
 280 PRINT AT 7,13;"b";AT 3,17;"
b";AT 6,24;"a"
 290 GO SUB 480
 300 PRINT AT 16,0;"DESTE MODO,P
ELO TEOREMA DE PI-  TAGORAS,TEMO
S..."
```

```
 310 PRINT AT 8,16;"b=SQR(4*r↑2-
a↑2)"
 320 GO SUB 480
 330 PRINT AT 16,0;"COMO A EXPRE
SSAO DA AREA,▦,E´"
 340 PRINT AT 10,20;"S=a*b"
 350 GO SUB 480
 360 PRINT AT 16,0;"SUBSTITUINDO
 O VALOR DE b NA  EXPRESSAO DA
AREA,VEM"
 370 PRINT FLASH 1;AT 8,16;"b";A
T 10,24;"b"
 380 GO SUB 480
 390 PRINT AT 12,14;"S=a*SQR(4*r
↑2-a↑2)"
 400 GO SUB 480
 410 PRINT AT 16,0;"A SUPERFICIE
,▦,E´ ENTAO..."
 420 PRINT AT 19,15;"S=";SQR (4*
r↑2-a↑2)
 430 GO SUB 480
 440 INPUT "OUTRO PROBLEMA?(S/N)
";a$
 450 IF a$="s" THEN CLS : GO TO
40
 460 CLS : PRINT AT 10,10;"ADEUS
": STOP
 470 STOP
 480 PRINT AT 21,7;"PRIMA UMA TE
CLA": PAUSE 0: PRINT AT 21,7;"
 490 FOR y=16 TO 19: FOR x=0 TO
31: PRINT AT y,x;" ": PAUSE 2: N
EXT x: NEXT y: RETURN
```

DESAFIO

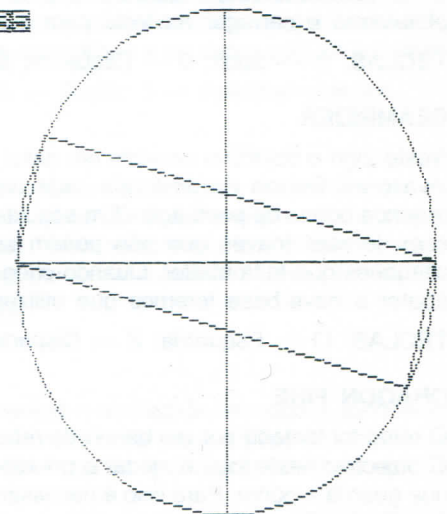
PROGRAMAÇÃO BASIC

Autor: R. C./PORTO

```
   5 REM    AREA DUM RECTANGULO
          INSCRITO - DEZ84
  10 PAPER 0: BORDER 0: INK 7: C
LS
  20 PRINT AT 1,0; BRIGHT 1;"CAL
CULO DA AREA DE UM RECTANGULO" '
"INSCRITO NUMA CIRCUNFERENCIA,DA
-"'"   DOS O RAIO E UM DOS LADO
S"
  30 INPUT BRIGHT 1;" RAIO   ";R
  35 IF R<=0 THEN BEEP .2,5: GO
TO 30
  40 INPUT BRIGHT 1;" LADO   ";L
  45 IF L<0 THEN BEEP .2,5: GO T
O 40
  50 IF L>2*R THEN BEEP .2,5: PR
INT INVERSE 1;''"    O LADO NAO P
ODE EXCEDER O           DIAM
ETRO           ": PAUSE 100: C
LS : GO TO 40
  60 PRINT ''"RAIO DA CIRCUNFER.
 -   ";R
  70 PRINT ''"LADO DO RECTANGULO
 -  ";L
  80 LET A=L*SQR (4*R↑2-L↑2)
  90 PRINT ''"AREA DO RECTANGULO
 -  ";A
 100 PRINT #0;TAB 7; INVERSE 1;"
PRIMA UMA TECLA ": PAUSE 0
 110 CLS
 120 LET E=87/R
 140 LET X=(L*E)↑2/(2*R*E)
 150 LET Y=SQR ((L*E)↑2-((L*E)↑4
/(4*(R*E)↑2)))
 200 CIRCLE 168,88,R*E
 210 PLOT 168-R*E,88: DRAW 2*R*E
,0
 220 PLOT 168,88-R*E: DRAW 0,2*R
*E
 230 PLOT 168-R*E,88
 240 DRAW X,Y
 250 DRAW 2*R*E-X,-Y
 260 DRAW -X,-Y
 270 DRAW -(2*R*E-X),Y
 300 PRINT INVERSE 1;"A=";A
 310 PRINT ' INVERSE 1;"L=";L
 330 PRINT ' INVERSE 1;"R=";R
 500 STOP
```

```
RAIO DA CIRCUNFER.   -   234.56

LADO DO RECTANGULO   -   123.45

AREA DO RECTANGULO   -   55871.685
```

```
CALCULO DA AREA DE UM RECTANGULO

INSCRITO NUMA CIRCUNFERENCIA,DA-

   DOS O RAIO E UM DOS LADOS
```

NOVOS PROGRAMAS

NOON WEEPER

Neste jogo, o objectivo consiste em abrir caminho para a nave poder viajar à procura de planetas, destruindo os meteoritos e quando um planeta se atravessar no trajecto da nave, ir de encontro a ele. Neste caso teremos um screen diferente, no qual o objectivo consistirá em destruir as máquinas ofensivas aliegenes e recolher os astronautas nesse planeta. Neste screen apenas podemos deslocar a nave para a esquerda e para a direita e disparar, enquanto que no 1 screen podemos também passar uma protecção em roda da nave, de modo a não sofer danos, em caso de colisão com os meteoritos.

1 SCREEN: 5 — Esquerda; 6 — Protecção; 8 — Direita; 0 — Disparo.
2 SCREEN: 5 — Esquerda; 0 — Disparo; 8 — Direita.

TURMOIL

Mais um jogo com a qualidade da «BUG BYTE SOFTWARE». Neste jogo o objectivo é recolher óleo do tanque existente em cada écran e nível e em seguida encher o carro de óleo, bastando para isso largar óleo através da grelha existente em todos os quadros. O programa tem 26 quadros diferentes, todos com o mesmo objectivo. Quando se acabar de encher um carro de óleo, passa-se automaticamente para outro quadro aumentando o grau de dificuldade; também é preciso evitar os árabes, e a nossa única defesa contra eles é deitar para o chão objectos e gotas de óleo, de modo que isso impeça que eles venham contra nós ou nos persigam.

TECLAS: Q — Subir; I — Esquerda: M — Largar óleo; Z — Descer; P — Direita.

PITFALL — Last caverns

Neste jogo você é um investigador e tem que perseguir as grutas e cavernas onde ficou preso, por causa de um desabamento das terras. Mas o sítio onde ficou preso é extraordinariamente perigoso: não só tem terríveis e eficientes armadilhas como também é povoado por seres igualmente perigosos e mortíferos . Por esse motivo este pesquisador terá que se movimentar com muito cuidado e precaução, pois à mínima falha da parte dele irá parar ao início das cavernas. Terá, no entanto, que recolher certos objectos de valor e outros que o poderão eventualmente ajudar a sair da sua prisão subterrânea. Evitar tocar nos animais ou objectos estranhos. Quando vir o rio pode nadar e mergulhar à procura de objectos. Para sair do rio basta encontrar uma plataforma e carregar na tecla para subir. Quando vir um obstáculo pode saltar por cima dele.

TECLAS: 2 — Subir; 0 — Esquerda; SPACE — Saltar; W — Descer; P — Direita.

BEAMRIDER

Neste jogo o objectivo consiste em deter as ondas de invasores que vão aparecendo no écran. Cada onda é constituída por 15 invasores. Depois de destruída cada onda aparece a nave-base no écran e temos que a tentar destruir. Se a destruirmos teremos bónus de pontuação. Em seguida aparecerá um écran igual ao anterior mas as ondas de invasores são mais completas, com defesas (naves que não podem ser destruídas). No centro esquerdo superior do écran aparece o número de naves aliegenes que falta abater. Quando chegar a 0 aparecerá a nave-base. No centro direito superior aparecerá a pontuação. Para abater a nave-base teremos que utilizar os mísseis-tecla 2.

TECLAS: O — Esquerda; Z — Dispara; P — Direita; 2 — Mísseis.

DRAGON FIRE

O reino foi tomado por um bando de rebeldes que, com o tesouro que possuem no castelo real, controlam toda a população.
O objectivo deste jogo é ajudar o príncipe a tomar o castelo e a recuperar o tesouro. Mas, primeiro, é preciso passar o dragão que guarda a ponte. Para isso é necessário evitar as labaredas que o dragão espele contra nós, baixando-se ou saltando. Para o 1 screen utilizar as seguintes teclas:

5 — Esquerda; 6 — Baixar-se; 8 — Direita; 0 — Saltar.

Este jogo é um autêntico desafio do computador à nossa perícia, reflexão e astúcia. Procure evitar o dragão da ponte, vencendo-o e conseguir os tesouros do príncipe. Boa sorte — Vai precisar dela!

STRANGELOOP

Jogo de aventura no qual o objectivo é evitar um determinado número de quadros (écrans) que nos permitirão apanhar certo número de objectos necessários à nossa sobrevivência. Com o programa seguem também instruções e um mapa em fotocópias.

DARTZS

Este jogo consiste no tradicional jogo de dardos, cujo objectivo consiste em atirar um dardo para um alvo fixo, tentando acertar no centro (mosca).
O programa fornece instruções pormenorizadas sobre o objectivo, modo de utilização, teclas e regras do jogo.
O jogo tem também a particularidade de poderem jogar 2 jogadores e de o computador fazer certas observações divertidas de quando em vez.
Boa sorte e . . . acerte na mosca!

THE INFERNO

Jogo de aventuras, um dos melhores da nova geração de aventuras para SPECTRUM.
Através de perguntas, frases, sugestões, etc., que o computador nos faz, devemos responder e dar-lhe instruções, tais como N, S, W, E, GO, KILL, etc...
O programa é acompanhado por fotocópias, que lhe darão melhores instruções e mais pormenores sobre a aventura.

WHITE LIGHTNING

WHITE LIGHTNING é o programa melhor e mais actualizado para fazer gráficos de alta resolução em alta velocidade, em 3 dimensões, etc... A alta definição dada por este programa a qualquer desenho, gráfico, «screen», etc., tornam este programa indispensável ao bom programador ou simples curioso em linguagem máquina. O programa é acompanhado de fotocópia com instruções detalhadas e mais explicações sobre o funcionamento do programa e sua aplicação.

PLUMMET

A nossa missão é libertar o ocupante do elevador evitando os guardas, evitando barreiras e armadilhas, bastando para isso subir aos andares, apanhar os objectos necessários e evitar os obstáculos. Podemos disparar contra os guardas. No 2 andar, não podemos passar pelas portas, se estas não estiverem abertas. Para as abrir basta disparar contra elas.

TECLAS: Q — Subir; O — Esquerda; A — Descer; P — Direita; SPACE — Disparar.

BUBBLE BUSHER

Neste jogo, comandamos o movimento de um boneco existente no fundo do écran, e cuja função é disparar contra as bolas que existem no écran. Sempre que uma bola é atingida ou choca contra o rasto do disparo, ela divide-se em duas partes. O objectivo é fazer sobreviver o boneco o mais tempo possível, passando de quadro sempre que se tenha acabado com as bolas. Os níveis superiores para os quais o boneco passa tem sempre um nível de dificuldade superior e uma bola mais.

TECLAS: A — Esquerda; D — Direita.

KNIGHT LORE

Trata-se de mais uma aventura com a qualidade da «BUG BYTE SOFT». Nós comandamos um pequeno boneco que se pode deslocar em todas as direcções e saltar, de modo que pode percorrer todas as salas, cavernas e divisões que constituem o jogo, assim como apanhar os objectos que for encontrando e evitar os perigos, armadilhas e seres inimigos. O objectivo é, como já foi dito, apanhar objectos e andar nas salas à procura dos mesmos. É preciso evitar as armadilhas e os seres que as guardam.

TECLAS: B — Rodar para a esquerda: Q — Saltar; N — Rodar para a direita; A — Andar; 1 — Apanhar objectos.

VOLCANO

Neste jogo, o objectivo é encontrar o barco de que necessitámos para fugir. Até o encontrarmos temos que evitar cair em armadilhas e tocar em objectos de modo a que a comida que transportámos para nos alimentar e dar energia não se esgote rapidamente. É preciso evitar ir de encontro a uma parede: perde-se comida. Perdemos 1 vida sempre que a comida se esgota. O jogo acaba, quando perdermos todas as vidas (5). Sugestão para o jogo e para percorrer o labirinto: sempre para NE.

TECLAS: 2 — Subir; 0 — Esquerda; SPACE — Saltar; W — Descer; P — Direita.

STAGE COACH

Neste jogo você é um cavaleiro (cowboy) e tem que saltar para a diligência, tomar as rédeas e conduzi-la de modo a apanhar os passageiros, conduzi-la dentro de cidades, evitar ataques de bandidos, de índios, etc... Para saltar para a diligência terá de se aproximar dela por detrás e, antes de lhe tocar, saltar, com a tecla de «FIRE». Se for a cavalo terá que ter cuidado com os cactos, destroços, etc..., pois se lhes tocar cairá da montanha.
TECLAS: As teclas podem ser definidas no programa. Para isso basta carregar em 1 e escolher: teclas definidas ou JOYSTICK.

NOTA: Cada vez que se termina um jogo, as teclas terão que ser novamente redefinidas.

CAVELON

O objectivo deste jogo é recolher todos os pedaços de porta, espalhados ao longo do labirinto, para passar os níveis seguintes. Ao passar o nível seguinte teremos que combater o feiticeiro. No labirinto, terá que evitar os arqueiros e tentar recolher o máximo de objectos. Quando lhe aparece a cruz terá que a apanhar, pois ela lhe dará o poder de destruir todos os arqueiros que se encontram nesse momento no écran. A destruição é obtida através da tecla «SYMBOL SHIFT». Se conseguir obter rapidamente todos os pedaços da porta, conseguirá um bom bónus. O bónus será tanto mais ou menos quanto mais rápida ou lentamente forem apanhados os pedaços da porta.

TECLAS: S — Subir; M — Esquerda; S.S. — Destruir.

HAVOC

O objectivo deste jogo é ultrapassar o maior número de aviões possível. O nosso avião pode-se mover para cima, para baixo, para a esquerda e para a direita. Pode-se também disparar. Como o cenário em que nos movemos está em perspectiva, temos que contar com o altímetro, existente no canto esquerdo inferior, e com a sombra do avião, existente no solo. Para se ultrapassar os obstáculos, terá que se ajustar uma determinada altura e depois move-se para a esquerda ou para direita, conforme for o

caso. Sempre que se encontra bidões, é conveniente disparar-se contra eles, visto que só deste modo conseguiremos o fuel que necessitamos para o avião.

TECLAS: Q — Subir; O — Esquerda; A — Descer; P — Direita; M — Disparar.

TWIN KINGDOM VALLEY

Este jogo consiste, mais exactamente, numa aventura dialogada. Mas tem também uma característica predominante: é a aventura dialogada com os melhores gráficos e figuras que existe actualmente no mercado. Com efeito, os gráficos são de alta resolução, usando os atributos pré-definidos do SPECTRUM, o que dá um efeito tri-dimensional a cada screen. Por outro lado, o diálogo é bastante bom, o computador aceita um número bastante bom de termos, valor, comandos, direcções, etc... Resumindo, tem um vocabulário com o suficiente para tornar esta aventura uma das mais atractivas existentes no mercado.
A história é-nos contada no próprio programa, assim como instruções pormemorizadas, vocabulário e comandos. Essencialmente, e bastante resumida, é a história de um vale onde existem 2 reis que, naturalmente, são rivais e inimigos.

POKER

É o tradional jogo de POKER, uma réplica muito bem conseguida da famosa máquina «GOOD LUCK» das Irmacor e Macofil. Partindo do princípio que toda a gente sabe jogar POKER neste tipo de máquina, será apenas necessário fornecer instruções quanto ao desenrolar do jogo em si. Deste modo, depois do jogo estar na memória do Spectrum, ele dá-nos 2 hipóteses:

I — Para instruções; P — Para jogar.

Quem desejar instruções sobre as teclas a utilizar, deverá carregar em «I». Quem já as souber deve carregar en «P». Quanto ao jogo, poderemos tentar dobrá-lo. Se o tentarmos dobrar, corremos o risco de perder tudo. O valor da aposta é dado no écran, assim como o valor atribuído a cada série, par, sequência, etc...
BOA SORTE.

PACMAN (Atarisoft)

É uma réplica quase perfeita da famosa máquina e com a qualidade que só a ATARISOFT nos poderia dar. Este jogo é o máximo que se pode exigir num jogo destes para o SPECTRUM: desde os sons, música perfeitos, passando pelos gráficos de alta resolução, é um jogo bastante perfeito em relação à máquina original. A nossa missão é comer todos os pontinhos existentes no labirinto, evitar os comilões que vêm atrás de nós, eventualmente, apanhar objectos ou frutos que fossem aparecer pelo labirinto. Se nos virmos aflitos com os comilões, bastará comermos uma cruz existente em cada canto (superior e inferior) do écran, e momentaneamente poderemos comer os comilões sem que eles nos façam dano algum. Sempre que o quadro esteja limpo de cruzes e pontos, passa-se para outro quadro, com um nível de dificuldade superior.

TECLAS: Q — Subir; O — Esquerda; A — Descer; P — Direita.

TRAVEL WITH TRASHAN

Este jogo é a continuação do jogo «TRASHMAN» que teve um grande sucesso, como jogo no Software do Spectrum. Esta versão, com melhoramento em relação à primeira, trata do mesmo assunto que a anterior: a recolha do lixo!
Só que desta vez a recolha não é feita apenas numa rua calma e solitária: é feita em diversos países do mundo. Assim para cada país o cenário para a recolha do lixo é típico do país em questão.
Ex.: Em França a recolha é feita numa esplanada dos «Champs Ellysées».
Boa recolha de lixo e boa sorte—Vai ser precisa.

NOTA: No jogo, sempre que queremos ir de viagem a um país, temos que pagar uma certa quantia em dinheiro. Esse dinheiro é conseguido por nós através da recolha de lixo nos vários países. Só assim é que podemos viajar. Sempre que escolhermos um país para trabalhar, depois da viagem, temos que aceitar um emprego que nos ofereçam. Só deste modo é que poderemos trabalhar neste país.

RIVER RAID

A nossa missão é destruirmos todas as fontes que encontrarmos, evitando assim que o inimigo invasor se desloque rapidamente. Assim, temos que patrulhar o rio, destruindo helicópteros, navios e fontes de abastecimento (fuel). Para abastecer o nosso meio de transporte, basta passar pelo «fuel» muito lentamente. O jogo tem níveis de dificuldade que são atingidos bastando para isso passar um certo número de fontes e destruí-las. Ex.: 5,20, 30,40, etc.
Por cada 10000 de pontuação, ganhámos um meio de transporte extra.

TECLAS: * Cursor ou; O — Esquerda; P — Direita; 2 — Desacelera; W — Acelera; M — Dispara.

* O programa é compatível com qualquer JOYSTICK.

INVADERS

A nossa missão é destruir as ondas sucessivas de aliens invasores que pretendem conquistar a terra. Para a sua defesa apenas existem 4 barreiras protectoras e 1 uma nave armada de laser, que se pode deslocar por baixo das barreiras. Quando um alien é atingido ele demora um bocado a desintegrar-se, mas já não nos poderá atingir.

TECLAS: 1 — Jogador; CAPS — Esquerda; Z — Direita; SPACE — Fire.
2 — Jogador; SYMBOL — Esquerda; SPACE — Direita; CAPS — Fire.

SPECIAL DELIVERY

O Pai Natal adormeceu e já só tem 5 horas para distribuir e recolher os presentes. O Pai Natal tem que recolher os presentes que os anjos lançam do céu e evitar chocar com as nuvens ou cumes das montangas, para não perder os presentes. Uma vez

recolhidos os presentes, o Pai Natal tem que passar com o seu trenó nos telhados das casas e descer pela chaminé — o pior são as chamas!
No fundo da chaminé, o Pai Natal dirige-se para a árvore a fim de colocar os presentes, evitando as crianças sonâmbulas. Tem então que encontrar a chave da porta para voltar ao trenó e prosseguir a distribuição.

TECLAS: 7 — Subir; 5 — Esquerda; 6 — Descer; 8 — Direita; 0 — Lançar presentes.

RIFLE RANGE

O objectivo deste jogo é disparar contra todos os animais na galeria de tiro para limpar um quadro. Mas cuidado, pois o número de disparos é limitado, o número de tiros de que se dispõe no início do nível do jogo em que jogamos. Assim, no 1 nível temos disparos no 2 e no 3. Cada animal atingido dar-nos-á 1 ponto e há 3 filas de 32 animais cada. Mas cuidado, se demoras muito tempo os esquilos começarão a reaparecer.
Para ganhar pontos e um número adicional de disparos, os prémios existentes no cimo do écran podem ser atingidos. Mas cuidado, se o dono da barraca estiver presente quando o prémio for atingido, o bónus de pontos será negativo. Mas se o dono não estiver presente então o bónus é positivo e conta para nós.

PONTUAÇÕES:

— Ursinho — mais 20 disparos; Peixinho dourado — mais 20 bónus; Embrulho — mais 10 bónus e mais 10 disparos.
Completado um écran, ganhamos um bónus de 20 pontos e 20 disparos extra, e proporciona-nos a possibilidade de jogar um jogo-bónus. Neste jogo teremos que acertar no maior número de alvos possível, de modo a ganhar o máximo de disparos extra possível para se usar no écran seguinte.

TECLAS: 6 — Esquerda; S — Começar o jogo; 7 — Direita; H — Parar o jogo; 0 — Disparo.

Disparando contra o dono da barraca (assassínio) resulta na sua prisão e consequente final de jogo. Mas se tiver batido uma «HIGH SCORE» não será levado em conta.
No nível 2 e 3 do jogo, os esquilos deixam cair bolotas e se atingirem a nossa arma perdemos alguns disparos.

EDUCAÇÃO MUSICAL

Este programa contém exercícios musicais, com arpejos, escalas, etc..., que o computador põe à disposição do utente. Por exemplo: o computador apresenta a pauta, demonstra a escala (DO,RE,...), executa um arpejo e pergunta se foi tocado em DO maior ou menor. Em seguida executa a escala outra vez, salienta uma determinada nota e pergunta qual foi e como se denomina, etc...

SURVIVAL

Este programa introduz os seus utilizadores nalguns dos princípios ecológicos (a interdependência dos seres vivos e o seu meio).
Os jogadores podem ver claramente as dificuldades que certas criaturas têm de enfrentar em lugares selvagens. Para sobreviverem e para se manterem alerta contra os inimigos, têm que encontrar alimentos e água suficientes.
A quantidade de alimentos necessários e probabilidade de serem mortos por um predador, dependem do habitat no qual o animal habita.
O programa ilustra o quanto são favoráveis ou desfavoráveis os habitats de diferentes espécies de animais.

DEUS EX MACHINA

Deus ex machina é um programa diferente de todos os programas que existem até agora no mercado de SOFTWARE para o Spectrum. Este excelente programa trata, como se fosse vídeo, toda a história, desde o nascimento do óvulo, a incubação do feto, até ao nascimento, os primeiros passos da criança, os seus movimentos, enfim: a história do homem desde que nasce até que morre.
O programa é mais para contemplar, pensar e observar o seu decor, do que propriamente para jogar. A nossa acção limita-se a pequenas intervenções ao longo do programa através de teclas já definidas. O programa está em duas partes. A 2.ª parte só pode ser «loadada» a partir do final da primeira parte.

MAKE-A-CHIP

Este programa permite-lhe explorar alguns aspectos do funcionamento do seu computador.
Um computador é constituído por um largo número de circuítos electrónicos.
MAKE-A-CHIP explicar-lhe-á acerca de circuitos electrónicos simples e elementos básicos do design do circuíto, e mostrar-lhe-á exemplos de alguns circuitos do ZX Spectrum.
Este programa poderá habilitá-lo a trabalhar com o design do circuito.
Poderá criar os seus próprios circuitos, combinando unidades básicas, e depois ver como os circuitos operariam.

CONTABILIDADE II — «POC»

O programa de contabilidade apresentado permite a mecanização de pequenas empresas, utilizando como base um microcomputador SINCLAIR «ZX-SPECTRUM» com 48 K e uma impressora 2040 também da TIMEX-SINCLAIR.
Pela 1.ª vez estes micros permitem, graças ao seu baixo custo, a automatização da gestão em empresas pequenas. O programa é um forte contributo para a utilização destes microcomputadores na área da gestão contabilista.
O programa está gravado em cassete ou em microdrive e a sua operação é de simplicidade extrema.

MERCADO Z80

O MERCADO Z80 É UMA SECÇÃO DO CLUBE Z80 QUE EMPRESA AOS SEUS SÓCIOS PROGRAMAS E LIVROS (SPECTRUM) PARA CONSULTA E MELHOR CONHECIMENTO/APROVEITAMENO DE MICROCOMPUTADORES.

— PROGRAMAS: Todos os jogos e utilitários que existem no CLUBE Z80, excepto programas de cópia e programas com direitos de autor (Ex.: "Cálculo de Estruturas").

— LIVROS: Cerca de 40 títulos diferentes.

COMO TORNAR-SE SÓCIO DO MERCADO Z80?

Para poder ter em sua posse 5 cassetes ou livros durante um mês, basta enviar-nos um depósito de Esc. 2000$00 (garantia de que os materiais nos serão devolvidos em estado de conservação e funcionamento idêntico àquele em que foram enviados).
Ao mesmo tempo, deverá remeter-nos a quantia de Esc. 1000$00 que será a base da sua "Conta-Corrente". Essa quantia servirá para pagar as suas despesas:

— Taxa de utilização dos produtos: 250$00 (referente a 5 unidades, entre livros e programas).

— Instruções dos programas (no caso de o sócio não as devolver, debitar-lhe-emos 5$00 por folha).

— Embalagem Postal: 20$00 a 30$00 (no caso de o pedido ser feito via CTT.

— Portes dos CTT's: 40$00 a 80$00 (no caso de o pedido ser feito via CTT.

IMPORTANTE!

— O depósito de 2000$00 pertence integralmente ao sócio desde que os materiais por ele utilizados nos sejam devolvidos nas mesmas condições em que saíram do CLUBE Z80. Assim, quando o sócio desistir do MERCADO Z80, essa quantia ser-lhe-á entregue.
Em caso de extravio, danos ou avarias dos materiais, o sócio pagará o valor comercial dos respectivos produtos (a descontar no depósito de 2000$00).

— No caso de os produtos seguirem via CTT, o sócio não pagará para levantar a encomenda. As despesas serão pagas por nós, no momento da expedição, e debitadas ao sócio (a descontar no depósito de 1000$00).

— Quando as suas despesas estiverem a atingir os 1000$00 avisá-lo-emos, e o sócio deverá renovar essa quantia de modo a cobrir despesas seguintes.

— A taxa de utilização dos produtos é fixa — 250$00. Ela refere-se ao conjunto de 5 unidades. (Pagará sempre 250$00 mesmo que peça só uma unidade).

QUE PRODUTO E QUE QUANTIDADES?

O sócio nunca pode pedir mais do que 5 unidades de cada vez (entre livros e cassetes). Quanto a livros não poderemos empresar mais do que um. Assim, o sócio poderá pedir:

— 5 cassettes
ou
— 4 cassettes + 1 livro

QUAL O TEMPO DE UTILIZAÇÃO?

O sócio poderá ficar com os produtos durante 1 MÊS, no máximo. Findo esse período, deverá devolvê-los ao CLUBE Z80.
O MERCADO Z80 só atenderá dois pedidos por mês, para cada sócio.

COMO FAZER O PEDIDO?

Numa carta, escreva pelo menos 10 títulos (por ordem de prioridade). Se os 5 primeiros não estiverem disponíveis, enviaremos os outros evitando grandes esperas de produtos que estejam em circulação.
Ao devolver os produtos, inclua uma carta com o pedido seguinte.

IMPORTANTE! O SÓCIO SÓ PODERÁ EFECTUAR UM NOVO PEDIDO JUNTAMENTE COM A DEVOLUÇÃO DO MATERIAL CORRESPONDENTE AO PEDIDO ANTERIOR (ou depois, se preferir).

Se estiver interessado no MERCADO Z80, faça já o seu 1.° pedido, enviando 3000$00 e o cupão abaixo devidamente preenchido.

INSCRIÇÃO NO MERCADO Z80

NOME ______

ENDEREÇO ______

CÓDIGO POSTAL ______

TELEFONE ______

ENVIO 3 000$00 (2 000$00 como garantia de que devolverei os produtos em boas condições + 1 000$00 para a minha «Conta Corrente» em:

☐ Cheque n.° ______ ☐ Vale Postal ☐ Dinheiro

Banco ______ N.° ______ ______

Data __/__/__ *Assinatura do Sócio* ______

É SÓCIO DO CLUBE Z80? ☐ SIM ☐ NÃO

CLUBE Z80

INSCRIÇÃO COMO ASSOCIADO

O **CLUBE Z80** está aberto a todos os utilizadores de microcomputadores.

A intenção de associar os entusiastas das micro-máquinas, é exclusivamente a de permitir:

1 — PUBLICAÇÃO DE UM JORNAL MENSAL, onde sejam publicados programas de uso geral ou específico como no caso da educação.

2 — PROMOVER TROCAS DE PROGRAMAS, e trocas de experiências; tanto no caso do Software (programação), como no caso do Hardware (electrónica).

3 — PROMOVER DESCONTOS NA AQUISIÇÃO DE PROGRAMAS.

4 — LANÇAR CURSOS DE PROGRAMAÇÃO EM BASIC — PASCAL OU OUTRAS LINGUAGENS E DIVULGAR O USO DE LINGUAGEM MÁQUINA.

NOME ....

IDADE .... COMPUTADOR TIPO ....

PROFISSÃO ....

ENDEREÇO ....

.... TELEF. ....

ASSINATURA ANUAL — Esc. 1 500$00 ☐

ASSINATURA SEMESTRAL — Esc. 750$00 ☐

CHEQUE OU VALE DO CORREIO

N.º ....

BANCO ....

DATA ..../......../........

JÁ SÓCIO ☐

NOVO SÓCIO ☐ → A partir do mês de .... (inclusive)